PATENT
Attorney Docket No.: 16869S-111200US
Client Ref. No.: W1470-01ES

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of:

HIROFUMI NAGASUKA *et al.*

Application No.: 10/804,942

Filed: May 18, 2004

For: STORAGE MANAGEMENT METHOD

Customer No.: 20350

Examiner: Unassigned

Technology Center/Art Unit: 2186

Confirmation No.: 8073

**PETITION TO MAKE SPECIAL FOR NEW APPLICATION UNDER M.P.E.P. § 708.02, VIII & 37 C.F.R. § 1.102(d)**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is a petition to make special the above-identified application under MPEP § 708.02, VIII & 37 C.F.R. § 1.102(d). The application has not received any examination by an Examiner.

(a) The Commissioner is authorized to charge the petition fee of $130 under 37 C.F.R. § 1.17(i) and any other fees associated with this paper to Deposit Account 20-1430.

03/02/2005 SMINASS1 00000004 10804942
01 FC:1464 130.00 DA

(b) All the claims are believed to be directed to a single invention. If the Office determines that all the claims presented are not obviously directed to a single invention, then Applicants will make an election without traverse as a prerequisite to the grant of special status.

(c) Pre-examination searches were made of U.S. issued patents, including a classification search, a computer database search, and a keyword search. The searches were performed on or around January 14, 2005, and were conducted by a professional search firm, Kramer & Amado, P.C. The classification search covered Class 711 (subclasses 162, 207, and 209) for the U.S. and foreign subclasses identified above. The computer database search was conducted on the USPTO systems EAST and WEST. The keyword search was conducted in Class 711 (subclasses 111, 148, 151, and 159). The inventors further provided a reference considered most closely related to the subject matter of the present application (see reference #4 below), which was cited in the Information Disclosure Statements filed on March 18, 2004.

(d) The following references, copies of which are attached herewith, are deemed most closely related to the subject matter encompassed by the claims:

(1) U.S. Patent No. 4,843,543;

(2) U.S. Patent No. 6,529,944 B1;

(3) U.S. Patent No. 4,985,828; and

(4) Japanese Patent Publication No. JP 2002-132552.

(e) Set forth below is a detailed discussion of references which points out with particularity how the claimed subject matter is distinguishable over the references.

A. Claimed Embodiments of the Present Invention

The claimed embodiments relate to a storage device management method in a computer system, a computer system, a storage device, and a management computer, particularly, to a method for managing a storage device in a computer system capable of effectively grasping the storage device use state and the file allocation information.

Independent claim 1 recites a storage management method for a computer system including a host computer, a storage sub-system having one or more volumes, a disk control unit provided in the storage sub-system for controlling operation of the storage sub-system, and a management computer for managing configuration of the storage sub-system. The method comprises registering, by the host computer, a volume identifier, a physical address, and a group identifier of the storage sub-system in a disk management table in the disk control units as a result of grouping of volumes in the storage sub-system according to a use purpose; obtaining, by the disk control unit, a physical address of a volume belonging to a same group as a group identifier input from the management computer from the disk management table, and notifying information described in a volume list of a volume allocated at the physical address to the management computer; and displaying, by the management computer, the information thus notified.

Independent claim 2 recites a storage management method for a computer system including a host computer, a storage sub-system having one or more volumes, a disk control unit provided in the storage sub-system for controlling operation of the storage sub-system, and a management computer for managing configuration of the storage sub-system. The method comprises registering, by the host computer, a volume identifier, a physical address, and a group identifier of the storage sub-system in the disk management table in a disk control unit as a result of grouping of volumes in the storage sub-system according to a use purpose; storing, by the disk control unit, information on the volume identifier and the physical address of the storage sub-system acquired at system starting in the disk management table, obtaining from the disk management table a physical address of a volume belonging to a same group as a group input from the management computer, and notifying information described in a volume list of a volume allocated at the physical address to the management computer; and displaying, by the management computer, the information thus notified.

Independent claim 5 recites a computer system comprising a host computer; a storage sub-system having one or more volumes and a disk control unit; and a management computer configured to manage configuration of the storage sub-system. The host computer transmits a volume identifier, a physical address, and a group identifier of the storage sub-system as a result of grouping of volumes in a storage sub-system according to a use purpose. The disk control unit obtains a physical address of a volume belonging to a same group as a

group transmitted from the management computer, and notifies information described in a volume list of a volume allocated at the physical address to the management computer. The management computer displays the information thus notified.

Independent claim 8 recites a storage system comprising a plurality of storage devices having one or more volumes; and a disk control unit connected to a host computer and a management computer configured to control the plurality of storage devices. The disk control unit further includes a disk management table for storing a volume identifier, a physical address, and a group identifier of a storage device as a result of grouping of volumes in the storage device according to a use purpose. The disk control unit obtains a physical address of a volume belonging to a same group as a group transmitted from the management computer, and notifies information described in a volume list of a volume allocated at the physical address to the management computer.

Independent claim 9 recites a management computer comprising a control section; and a storage device. The control section comprises a plurality of pairs of one or more volumes and a disk control unit for managing configuration of storage used by a host computer. Information described in a disk management table in a disk control unit transmitted from disk control units is stored together with identifiers of the disk control units in the storage device, a disk control unit to which a volume corresponding to a requested group or a requested volume identifier belongs is requested to acquire information on a volume list concerning the volume belonging to the group or volume thus requested, and according to information on a volume list as a response to this request, an information file of the disk management table stored in the storage device is updated to the disk management table information thus notified.

Independent claim 10 recites a management program stored in a storage medium and executed in a management computer including a plurality of pairs of one or more volumes and a disk control device for managing configuration of a storage sub-system used by a host computer. The program comprises code for storing information described on a disk management table in a disk control unit from respective disk control units together with identifiers of the disk control units into a storage device accessible by the management computer; code for requesting the disk control unit to which a volume corresponding to a requested group or a requested volume identifier belongs, to acquire volume list information concerning a volume belonging to the group or the volume identifier thus requested; and code

for, according to the volume list information as a response to the request, updating an information file in the disk management table stored in the storage device accessible by the management computer to the disk management table information thus notified.

Independent claim 11 recites a computer system comprising a host computer; a storage sub-system having one or more volumes and a disk control unit; and a management computer configured to manage configuration of the storage sub-system. The host computer transmits a volume identifier, a physical address, and a group identifier of the storage sub-system as a result of grouping of volumes in the storage sub-system according to a use purpose. The disk control unit obtains a physical address of a volume belonging to a same group as a group transmitted from the management computer, and notifies information described in a volume list of a volume allocated at the physical address to the management computer. The management computer displays the information thus notified. The management computer stores information described in the disk management table together with a disk control unit identifier from respective disk control units to a storage device accessible by the management computer, and requests a disk control unit to which a volume corresponding to a requested group or a requested identifier belongs to acquire volume list information concerning the volume belonging to the group or the volume identifier thus requested. The disk control unit which has received the request notifies information described in a volume list of a volume identical to the group or the volume identifier specified by the management computer to the management computer. Each of the control units notifies information requested to be acquired from the management computer to the management computer, and notifies new information on the disk management table to the management computer. The management table thus notified updates an information file of the disk management table stored in the storage device accessible by the management computer to the disk management table thus notified.

One of the benefits that may be derived is that it is possible to effectively acquire the storage device information such as information on the storage device use condition and information on the files stored without increasing the load on the host computer, and it is possible to acquire information for managing the large-scale storage system use condition without increasing the load on the host computer.

B. Discussion of the References

1. U.S. Patent No. 4,843,543

This reference relates to a storage control method and apparatus with a plurality of identifier transmission means for dividing the access requests issued from the plurality of access request control units into a plurality of groups based on the order of issuance from the access request control units. Each identifier transmission means is associated with one of the access request units and divides the access requests from the associated access request unit. Each identifier transmission means further adds access requests in each group with access request identifiers.

The reference does not teach registering, by the host computer, a volume identifier, a physical address, and a group identifier of the storage sub-system in a disk management table in the disk control units as a result of grouping of volumes in the storage sub-system according to a use purpose; obtaining, by the disk control unit, a physical address of a volume belonging to a same group as a group identifier input from the management computer from the disk management table, and notifying information described in a volume list of a volume allocated at the physical address to the management computer, as recited in independent claims 1, 2, 5, and 8. Nor does it disclose storing information described on a disk management table in a disk control unit from respective disk control units together with identifiers of the disk control units into a storage device accessible by the management computer; requesting the disk control unit to which a volume corresponding to a requested group or a requested volume identifier belongs, to acquire volume list information concerning a volume belonging to the group or the volume identifier thus requested; and according to the volume list information as a response to the request, updating an information file in the disk management table stored in the storage device accessible by the management computer to the disk management table information thus notified, as recited in independent claims 9, 10, and 11.

2. U.S. Patent No. 6,529,944 B1

This reference discloses a host system for remote control of mass storage volumes using cascading commands with a selected identifier being in the command as issued from the host system. The command is recognizable by each mass storage system in

the stream of mass storage systems. The selected identifier is selected from a group of identifiers. The identifiers in the group identify the mass storage systems in the stream of mass storage systems.

The reference does not teach registering, by the host computer, a volume identifier, a physical address, and a group identifier of the storage sub-system in a disk management table in the disk control units as a result of grouping of volumes in the storage sub-system according to a use purpose; obtaining, by the disk control unit, a physical address of a volume belonging to a same group as a group identifier input from the management computer from the disk management table, and notifying information described in a volume list of a volume allocated at the physical address to the management computer, as recited in independent claims 1, 2, 5, and 8. Nor does it disclose storing information described on a disk management table in a disk control unit from respective disk control units together with identifiers of the disk control units into a storage device accessible by the management computer; requesting the disk control unit to which a volume corresponding to a requested group or a requested volume identifier belongs, to acquire volume list information concerning a volume belonging to the group or the volume identifier thus requested; and according to the volume list information as a response to the request, updating an information file in the disk management table stored in the storage device accessible by the management computer to the disk management table information thus notified, as recited in independent claims 9, 10, and 11.

3. U.S. Patent No. 4,985,828

This reference discloses a method and apparatus for generating a real address, multiple virtual address spaces of a storage provides group identifiers in each of the entries in the address translation table, the address translation buffer and the entry in the address control register for identifying a respective area shared by a specific group of virtual spaces. See column 3, lines 12-67.

The reference does not teach registering, by the host computer, a volume identifier, a physical address, and a group identifier of the storage sub-system in a disk management table in the disk control units as a result of grouping of volumes in the storage sub-system according to a use purpose; obtaining, by the disk control unit, a physical address

of a volume belonging to a same group as a group identifier input from the management computer from the disk management table, and notifying information described in a volume list of a volume allocated at the physical address to the management computer, as recited in independent claims 1, 2, 5, and 8. Nor does it disclose storing information described on a disk management table in a disk control unit from respective disk control units together with identifiers of the disk control units into a storage device accessible by the management computer; requesting the disk control unit to which a volume corresponding to a requested group or a requested volume identifier belongs, to acquire volume list information concerning a volume belonging to the group or the volume identifier thus requested; and according to the volume list information as a response to the request, updating an information file in the disk management table stored in the storage device accessible by the management computer to the disk management table information thus notified, as recited in independent claims 9, 10, and 11.

4. Japanese Patent Publication No. JP 2002-132552

This reference relates to an information processing system capable of specifying not only data set name but also physical position, volume catalog only and volume name or data set as the units to set and release resident on a cache memory. In the system, a utility program 1 of resident control enables stay resident with issuing information of the physical position necessary to release resident to a disk array subsystem 13. For the specification of the volume catalog/volume name, the program 1 acquires physical information necessary to release resident from volume catalog information in a volume 14 in a disk-driving device 9. The program 1 acquires the physical position necessary to release resident by comparing resident control information 7 of dynamic cache resident mechanism with information of the volume catalog to delete spaces remaining on the cache memory.

According to this conventional technique, a cache memory provided in the disk control unit of the storage device stores a volume list and a catalogue file as file allocation information. Moreover, physical region data is resident in the cache memory and the resident state can be released. The volume list is stored in the cache memory, which is not accompanied by the I/O operation to/from the disk device of the storage device and accordingly, it becomes possible to acquire the volume list information with low overheads. Moreover, since the catalogue managing the entire system file information can be handled as

a type of file, the technique can acquire the catalog information with low overheads like the volume list. Present application at page 1, line 21 to page 2, line 12.

This conventional technique has no consideration on increase of the overheads required for information acquisition due to increase of the number of devices accompanying the increase of the computer system size. Since the volume list exists on a storage device basis, a problem arises that as the number of storage devices used increases, the overheads required for information acquisition increase. Especially in this conventional technique, information is acquired via a host computer and there may arise a problem of deteriorating the throughput or response time of work executed in the host computer. The catalog file also has a problem that increase of the overhead required for information acquisition accompanying the catalogue file size is expected as the system size increases. Moreover, as the system size increases, not only the catalog for managing the entire computer system is generated but also a great number of catalog files are generated according to the work and use purpose. The technique has a problem that such an increase of the number of catalog files also causes the increase of overheads required for information acquisition. Present application, at page 2, line 15 to page 3, line 12.

The reference does not teach registering, by the host computer, a volume identifier, a physical address, and a group identifier of the storage sub-system in a disk management table in the disk control units as a result of grouping of volumes in the storage sub-system according to a use purpose; obtaining, by the disk control unit, a physical address of a volume belonging to a same group as a group identifier input from the management computer from the disk management table, and notifying information described in a volume list of a volume allocated at the physical address to the management computer, as recited in independent claims 1, 2, 5, and 8. Nor does it disclose storing information described on a disk management table in a disk control unit from respective disk control units together with identifiers of the disk control units into a storage device accessible by the management computer; requesting the disk control unit to which a volume corresponding to a requested group or a requested volume identifier belongs, to acquire volume list information concerning a volume belonging to the group or the volume identifier thus requested; and according to the volume list information as a response to the request, updating an information file in the disk management table stored in the storage device accessible by the management computer to the

disk management table information thus notified, as recited in independent claims 9, 10, and 11.

(f) In view of this petition, the Examiner is respectfully requested to issue a first Office Action at an early date.

Respectfully submitted,

Chun-Pok Leung
Reg. No. 41,405

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 650-326-2400
Fax: 415-576-0300
Attachments
RL:rl
60415574 v1

W1470

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-132552
(43)Date of publication of application : 10.05.2002

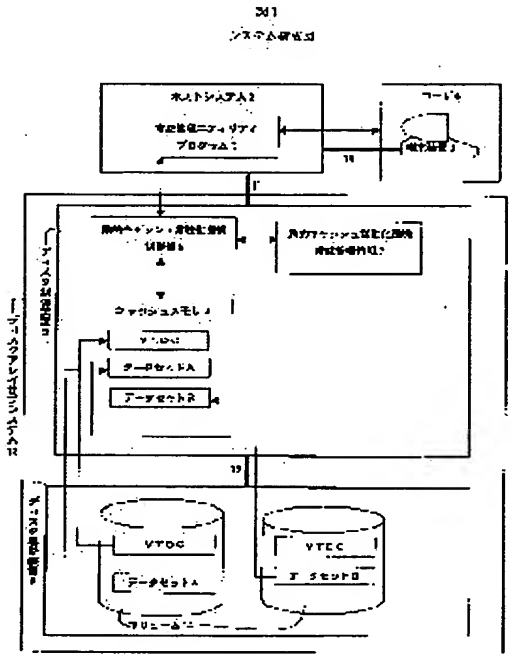
(51)Int.Cl. G06F 12/00
G06F 3/06
G06F 12/08
G06F 12/12
G06F 13/10

(21)Application number : 2000-327009
(22)Date of filing : 20.10.2000

(71)Applicant : HITACHI LTD
(72)Inventor : NASHIMOTO KUNIHIKO
YANAKA MASARU

## (54) INFORMATION PROCESSING SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide an information processing system capable of specifying not only data set name but also physical position, volume catalog only and volume name or data set as the units to set and release resident on a cache memory.
SOLUTION: In the system, a utility program 1 of resident control enables stay resident with issuing information of the physical position necessary to release resident to a disk array sub system 13. For the specification of the volume catalog/volume name, the program 1 acquires physical information necessary to release resident from volume catalog information in a volume 14 in a disk-driving device 9. The program 1 acquires the physical position necessary to release resident by comparing resident control information 7 of dynamic cache resident mechanism with information of the volume catalog to delete spaces remaining on the cache memory.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

W1470

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－132552
（P2002－132552A）
(43)公開日　平成14年５月10日(2002.5.10)

| (51)Int.Cl.⁷ | 識別記号 | FI | テーマコード（参考） |
|---|---|---|---|
| G06F 12/00 | 514 | G06F 12/00 | 514K 5B005 |
| | 520 | | 520J 5B014 |
| 3/06 | 302 | 3/06 | 302A 5B065 |
| | 540 | | 540 5B082 |
| 12/08 | 557 | 12/08 | 557 |

審査請求　未請求　請求項の数４　ＯＬ　（全 10 頁）　最終頁に続く

(21)出願番号　特願2000－327009(P2000－327009)
(22)出願日　平成12年10月20日(2000.10.20)

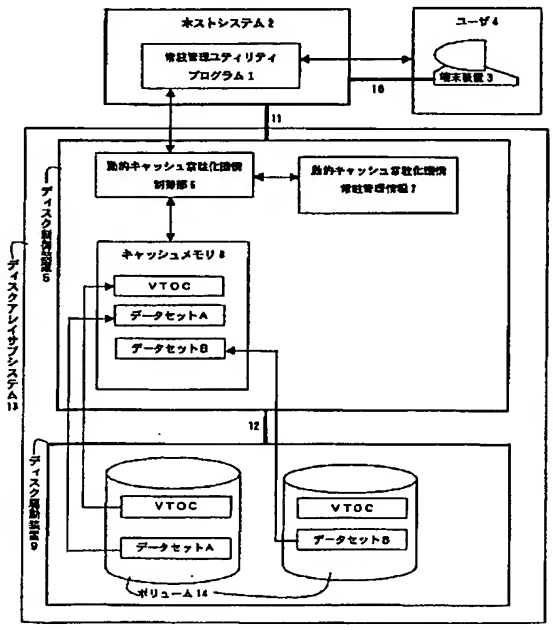
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　梨本　国彦
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内
(72)発明者　谷中　大
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内
(74)代理人　100075096
弁理士　作田　康夫

最終頁に続く

(54)【発明の名称】　情報処理システム

(57)【要約】
【課題】キャッシュメモリ上に常駐化設定および常駐化解除する際の指定の単位が、データセット名だけに限定されず、物理的な位置の指定またはボリューム目録のみの指定、ボリューム名、またはデータセット指定が可能になる情報処理システムを提供する。
【解決手段】常駐管理ユティリティプログラム１が、常駐解除に必要である物理的な位置の情報をディスクアレイサブシステム１３に発行することにより常駐を可能とする。ボリューム目録／ボリューム指定は、常駐管理ユティリティプログラム１がディスク駆動装置９内のボリューム１４中にあるボリューム目録情報から、常駐解除に必要である物理的な情報を得る。また、常駐管理ユティリティプログラム１は、動的キャッシュ常駐化機構常駐管理情報７とボリューム目録の情報とを比較することにより、常駐解除に必要である物理的な位置を取得し、キャッシュメモリ上に残留する領域を削除する。

図１
システム構成図



【特許請求の範囲】
【請求項１】データを一時格納するキャッシュメモリと、データの書き込み及び読み出しを行う上位装置を接続するインタフェースとを備えた外部記憶装置であって、
上位装置からのコマンドによって前記キャッシュメモリ上にユーザデータをボリューム単位、データセット単位、ボリューム目録（ＶＴＯＣ）、又は物理的な位置で指定して常駐化又は常駐化解除を行うことを特徴とする外部記憶装置。
【請求項２】データを一時格納するキャッシュメモリと、データの書き込み及び読み出しを行う上位装置を接続するインタフェースとを備えた外部記憶装置であって、
前記キャッシュメモリ上に常駐しているユーザデータが前記外部記憶装置上から削除された場合に、前記キャッシュメモリ上に常駐しているデータ領域を開放することを特徴とする外部記憶装置。
【請求項３】データを一時格納するキャッシュメモリを具備した外部記憶装置と、この外部記憶装置に接続されこの外部記憶装置に対するデータの書き込み及び読み出しを行うホストシステムとを備えた情報処理システムであって、
前記ホストシステムは、前記キャッシュメモリ上のデータを管理するユティリティプログラムと、このユティリティプログラムが動作するために必要な管理情報を備え、ユティリティプログラムは前記外部記憶装置の状態情報を備え、前記外部記憶装置は前記管理情報及び前記状態情報に基づいた命令を受領し、ユーザデータのキャッシュメモリ上への常駐及び常駐解除の指示単位として、ボリューム単位、データセット単位、ボリューム目録（ＶＴＯＣ）、物理的な位置で指定することを特徴とする情報処理システム。
【請求項４】データを一時格納するキャッシュメモリを具備した外部記憶装置と、この外部記憶装置に接続されこの外部記憶装置に対するデータの書き込み及び読み出しを行うホストシステムとを備えた情報処理システムであって、
前記ホストシステムは、前記キャッシュメモリ上のデータを管理するユティリティプログラムと、このユティリティプログラムが動作するために必要な管理情報を備え、ユティリティプログラムは前記外部記憶装置の状態情報を備え、前記外部記憶装置は前記管理情報及び前記状態情報に基づいた命令を受領し、データを前記キャッシュメモリ上に常駐または前記キャッシュメモリから常駐解除を行い、
前記キャッシュメモリ上に常駐しているデータセットが前記外部記憶装置上から削除された場合に、前記キャッシュメモリ上に常駐しているデータ領域を開放する手段を備えたことを特徴とする情報処理システム。

【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、キャッシュメモリを具備した外部記憶装置のキャッシュメモリ制御技術に関し、特にキャッシュメモリ上への情報の常駐設定あるいは常駐解除を行う指示単位の制御技術に関する。
【０００２】
【従来の技術】近年、情報処理システムの高性能・大規模化に伴い、外部記憶装置においても、より一層の高性能が求められている。特に外部記憶装置においては、制御装置に大容量キャッシュメモリの効率の良い制御方法が重要となってきている。このため、大容量キャッシュメモリの制御に関する方式が、以下の様に考案されている。
【０００３】特開平７－２８７６６９号公報には、キャッシュメモリ上に配置すべき主記憶上のデータをプログラムが明示的に指令可能な命令と、指定に従って主記憶上のデータをキャッシュメモリに転送する制御回路と、転送されたデータをキャッシュメモリ上に常駐化させる回路と、常駐化されたデータの常駐化を無効にする回路とを設け、処理を優先したい特定のプログラムが、その処理手順の特性に合わせてキャッシィングを制御する方式が開示されている。
【０００４】
【発明が解決しようとする課題】キャッシュメモリ制御については、出願人により特願平１１－１９９７６２号として、データセット名の入力によるユティリプログラムを用いて、複数トラックのユーザデータを外部記憶装置のキャッシュメモリ上に常駐化設定および常駐化解除の操作をホストシステムにより行なうキャッシュ制御と、外部記憶装置が稼動情報を採取することにより、ホストシステムの稼動状況に応じてキャッシュメモリ上に常駐化および常駐化解除をホストシステムにより行ない情報処理システムの性能向上を行なうキャッシュ制御の方式が出願されている。
【０００５】特願平１１－１９９７６２号により出願された技術では、ホストシステムに組み込まれたユティリティプログラムにより、ユーザデータをデータセット指定でキャッシュメモリ上に常駐設定及び、常駐解除することができるが、データの物理的な位置の指定、またはボリューム目録（ＶＴＯＣ：Volume table of contents）のみの指定、ボリューム名指定、データセットの常駐化設定および常駐解除ができない。
【０００６】また、特願平１１－１９９７６２号で出願された技術では、ホストシステムに組み込まれたユティリティプログラムによりデータをキャッシュメモリに常駐化設定した後、キャッシュメモリに常駐化設定したデータセットを常駐化解除せずに削除した場合、キャッシュメモリ上に削除したデータセットの領域定義が残留してしまう。キャッシュメモリ上に残留している領域は開

放（常駐解除）しないと、他のデータを格納することができない。なぜならば、ホストシステムがデータセットを削除した事象を、外部記憶装置で能動的に知る方法が無いためである。

【０００７】本発明の目的は、キャッシュメモリ上に常駐化設定および、常駐化解除する際の指定の単位がデータセット名だけに限定されず、物理的な位置の指定、またはボリューム目録（ＶＴ0Ｃ）のみの指定、ボリューム名、またはデータセット指定が可能になる、情報処理システムを提供することにある。

【０００８】本発明の他の目的は、キャッシュメモリ上に常駐化設定したデータセットが削除された場合、キャッシュメモリ上に残留した領域を解除するキャッシュ制御技術を持った情報処理システムを提供することにある。

【０００９】

【課題を解決するための手段】前述の目的を達成する為には、以下の手段を実施する。

【００１０】まず、キャッシュメモリ上に常駐化設定および、常駐化解除する際の指定の単位が物理的な位置での指定、ボリューム目録（ＶＯＴＣ）領域のみの指定、または、データセット指定の場合を説明する。ユーザが物理的な位置で常駐指定を行う場合は、ユティリティプログラムを通じて外部記憶装置に物理的な位置の情報で常駐指示命令を発行する。

【００１１】次に、ユーザがボリューム目録（ＶＴＯＣ）領域、またはデータセット単位または、ボリューム単位を指定して常駐化設定及び常駐化解除指定を行う場合は、ユティリティプログラムが、ホストシステムが管理しているシステム管理情報内のボリューム情報から、ボリューム目録（ＶＴＯＣ）領域または、ボリューム中のデータセットが存在する領域または、ボリューム中のデータが存在する領域の物理的な位置を取得し、取得した物理的な位置で常駐指示命令を外部記憶装置に発行する。外部記憶装置は、ユティリティプログラムから命令された物理的な位置領域を内部メモリに格納する。

【００１２】前述のほかの目的を達成するためには、以下の手段を実施する。キャッシュメモリ上に常駐化設定したデータを削除したことによりキャッシュメモリ上にデータが存在しない領域が残留する。このキャッシュメモリ上に残留した領域を常駐解除する方法を説明をする。

【００１３】ユティリティプログラムは、キャッシュメモリ上に常駐しているデータの物理的な位置を、外部記憶装置内の常駐管理情報から取得できる。また、現在ボリュームに存在しているデータセットの情報はホストシステムが提供しているシステム管理情報で取得することができる。ユティリティプログラムでは前記２つの情報を比較し、現在存在していないデータセットで、且つキャッシュメモリ上に存在(残留)している領域を探し出し、常駐解除指示命令を外部記憶装置へ発行する。

【００１４】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照しながら詳細に説明する。図１は、本発明の実施の形態であるユティリティプログラム（以下、常駐管理ユティリティプログラム１と言う）を含む情報処理システムの構成図である。以下で、情報処理システムの制御の流れを説明する。

【００１５】ユーザ４は端末装置３を使用し、ケーブル１０を通じてホストシステム２の中央処理装置に組み込まれている常駐管理ユティリティプログラム１に常駐指示をする。

【００１６】常駐管理ユティリティプログラム１は、ケーブル１１を通じて外部記憶装置（以下ディスクアレイサブシステム１３と呼ぶ）内のディスク制御装置５に設けられた動的キャッシュ常駐化機構制御部６に対して、常駐指示命令を発行する。

【００１７】ディスクアレイサブシステム１３は、ディスク制御装置５とディスク駆動装置９から構成されいる。

【００１８】動的キャッシュ常駐化機構制御部６は、ホストシステム２に組み込まれている常駐管理ユティリティプログラム１からの指示に従いキャッシュメモリ８へディスク駆動装置９内のデータセット又はボリューム目録（ＶＴＯＣ）を常駐し、キャッシュメモリ８上の常駐情報を、動的キャッシュ常駐化機構常駐管理情報７に記録する。

【００１９】ディスク制御装置５の構成要素として、動的キャッシュ常駐化機構制御部６、動的キャッシュ常駐化機構常駐管理情報７とキャッシュメモリ８があり、さらにディスク駆動装置９を制御する。

【００２０】ディスク駆動装置９は、ホストシステム２から書かれたデータを保持している。

【００２１】動的キャッシュ常駐化機構制御部６は、ホストシステム２からの常駐化設定及び常駐化解除命令の処理、キャッシュメモリ８とのデータ転送処理、動的キャッシュ常駐化機構常駐管理情報７とのデータ転送処理を行なう。

【００２２】動的キャッシュ常駐化機構常駐管理情報７は、動的キャッシュ常駐化機構制御部６で使用する情報が格納されている。

【００２３】キャッシュメモリ８はホストシステム２からのライトデータ及び、ディスク駆動装置９からのリードデータを一時格納しておくメモリである。

【００２４】ボリューム１４は、ディスク駆動装置を構成している記憶媒体である。

【００２５】次に、常駐管理ユティリティプログラムが常駐化設定行なう時の制御の流れを図２を参照して説明する。図２は本発明の実施の形態である動的キャッシュ常駐化の制御を実現するための手段であるホストユティ

リティプログラムが、キャッシュ常駐化設定を指示した場合の制御の流れを示している。

【００２６】ユーザ４はステップ１５で、ボリューム目録（ＶＴＯＣ）領域指定または、データセット名指定での常駐化設定指示を行なう。

【００２７】ボリューム目録（ＶＴＯＣ）は、データセットの管理情報、データセット名、エクステント情報、データセットの属性の情報を記録している。ボリューム目録（ＶＴＯＣ）は、ボリューム１４中の一部の領域を設けて存在する。

【００２８】エクステント情報は、ディスク駆動装置９内のボリューム１４中のデータセット格納領域の物理的な位置を示している情報である。

【００２９】ステップ１７で、引き渡されたボリューム目録（ＶＴＯＣ）領域の指定、またはボリューム指定で常駐化設定指示のコマンドパラメータを取得する。

【００３０】ステップ１８で、常駐管理ユティリティプログラム１は、ディスク駆動装置９内のボリューム１４中のボリューム目録（ＶＴＯＣ）情報を取得する。

【００３１】ステップ１９では、ステップ１８で取得したボリューム目録（ＶＴＯＣ）情報のエクステント情報より、ボリューム１４中のボリューム目録（ＶＴＯＣ）が存在する領域または、ボリューム１４中のデータセットが存在する領域の始まりと終わりの物理的な位置の情報と、ステップ１７で、コマンドパラメータとして取得したボリューム目録（ＶＴＯＣ）領域の指定、またはデータセット名指定の情報を比較する。

【００３２】ステップ２０で常駐化設定指示した領域に該当する物理的な位置を取得する。

【００３３】ステップ２１では、ステップ２０で受け取った物理的な位置での領域で常駐化設定する指示を動的キャッシュ常駐化機構の制御部６に発行する。

【００３４】ステップ２３で、常駐管理ユティリティプログラム１から受け取った常駐化設定指示した物理的な位置の領域に該当するディスク駆動装置９のデータをキャッシュメモリ上に常駐化設定する。

【００３５】ステップ２４では、ステップ２３によりキャッシュメモリ上に常駐化設定したデータの物理的な位置の領域を動的キャッシュ常駐化機構常駐管理情報７に記録する。

【００３６】ステップ２５で、動的キャッシュ常駐化機構制御部６は常駐化設定の処理を終了する。ステップ２２で、常駐管理ユティリティプログラム１は、常駐化設定の処理の終了報告を受け取り終了する。

【００３７】また、ユーザ４が物理的な位置で常駐化設定する場合（ステップ１６）、ボリューム１４中のボリューム目録（ＶＴＯＣ）との比較（ステップ１９）を行なわない為、ユーザ４で、入力された（ステップ１６）の物理的な位置を元に、常駐化設定指示を動的キャッシュ常駐化機構制御部６へ発行し（ステップ２１）、動的キャッシュ常駐化機構制御部６は、常駐化設定処理（ステップ２３～２５）を行なう。

【００３８】次に、常駐管理ユティリティプログラムが常駐化解除行なう時の制御の流れを図３を参照して説明する。図３は本発明の一実施の形態である動的キャッシュ常駐化の制御を実現するための手段であるホストユティリティプログラムが、キャッシュ常駐化解除を指示した場合の制御の流れを示している。

【００３９】ユーザ４はステップ２６で、ボリューム目録（ＶＴＯＣ）領域の指定または、ボリューム名指定または、データセット名指定での常駐化解除指示を行なう。

【００４０】ステップ２８で、引き渡されたボリューム目録（ＶＴＯＣ）領域の指定または、ボリューム指定または、データセット名指定で常駐化解除指示のコマンドパラメータを取得する。

【００４１】ステップ２９で、常駐管理ユティリティプログラム１は、ディスク駆動装置９内のボリューム１４中のボリューム目録（ＶＴＯＣ）情報を取得する。

【００４２】ステップ３０では、ステップ２８で取得したボリューム目録（ＶＴＯＣ）情報のエクステント情報であるボリューム１４中のボリューム目録（ＶＴＯＣ）が存在する領域または、ボリューム１４中のデータが存在する領域、またはボリューム１４中のデータセットが存在する領域の始まりと終わりの物理的な位置の情報と、ステップ２８でコマンドパラメータとして取得したボリューム目録（ＶＴＯＣ）領域の指定または、ボリューム指定または、データセット名指定の情報を比較する。

【００４３】ステップ３１で、常駐化解除指示した領域に該当する物理的な位置を取得する。

【００４４】ステップ３２では、ステップ３１で受け取った物理的な位置の領域を常駐化解除する指示を動的キャッシュ常駐化機構の制御部６に発行する。

【００４５】ステップ３４で、常駐管理ユティリティプログラム１から受け取った常駐化解除指示した物理的な位置の領域に該当するディスク駆動装置９のデータをキャッシュメモリ上から常駐化解除する。

【００４６】ステップ３５では、ステップ３４によりキャッシュメモリ上から常駐化解除したデータの領域の物理的な位置を動的キャッシュ常駐化機構常駐管理情報７から削除する。

【００４７】ステップ３６で、動的キャッシュ常駐化機構制御部６は常駐化解除の処理を終了する。

【００４８】ステップ３３で、常駐管理ユティリティプログラム１は、常駐化解除の処理の終了報告を受け取り終了する。

【００４９】また、ユーザ４が物理的な位置で常駐化解除する場合（ステップ２７）、ボリューム１４中のボリューム目録（ＶＴＯＣ）との比較（ステップ３０）を行

なわない為、ユーザ４で、入力された（ステップ２７）の物理的な位置を元に、常駐化解除指示を動的キャッシュ常駐化機構制御部６へ発行し（ステップ３２）、動的キャッシュ常駐化機構制御部６は、常駐化解除処理（ステップ３４～３６）を行なう。

【００５０】次に、常駐管理ユティリティプログラムが常駐化設定しているデータセットを削除した場合に、キャッシュメモリ上に常駐している領域の常駐化解除を行なう時の制御の流れを図４を参照して説明する。図４は本発明の一実施の形態である動的キャッシュ常駐化の制御を実現するための手段であるホストユティリティプログラムが、常駐化設定したデータセットを削除した場合、キャッシュ常駐化解除を指示した場合の制御の流れを示している。

【００５１】キャッシュメモリ上に常駐化設定したデータセットを削除した場合、キャッシュメモリ上に削除後もデータセットの情報を格納していた領域が残留する。

【００５２】ユーザ４は、ステップ３７で、常駐化解除指示を行なう。

【００５３】ステップ３８で、常駐管理ユティリティプログラム１は、ボリューム目録（ＶＴＯＣ）情報を取得する。

【００５４】ステップ３９で、常駐管理ユティリティプログラム１は動的キャッシュ常駐化機構の常駐管理情報７の取得命令を動的キャッシュ常駐化機構制御部６に発行する。

【００５５】ステップ４４で、動的キャッシュ常駐化機構の制御部６は、動的キャッシュ常駐化機構常駐管理情報７を取得し、常駐管理ユティリティプログラム１に返す。

【００５６】ステップ４０で、ステップ３８で取得したボリューム目録（ＶＴＯＣ）情報のエクステント情報によるデータセットが存在する領域の物理的な位置と、動的キャッシュ常駐化機構の常駐管理情報７による常駐している物理的な位置を比較する。

【００５７】ステップ４１で、ステップ４０の比較の結果からデータセットが存在しない物理的な位置を取得する。

【００５８】ステップ４２では、ステップ４１で受け取った物理的な位置の領域を常駐化解除する指示を動的キャッシュ常駐化機構の制御部６に発行する。

【００５９】ステップ４５で、常駐管理ユティリティプログラム１から受け取った常駐化解除指示された物理的な位置に該当するディスク駆動装置９のデータをキャッシュメモリ上から常駐化解除する。

【００６０】ステップ４６では、ステップ４５によりキャッシュメモリ上から常駐化解除したデータの領域の物理的な位置を動的キャッシュ常駐化機構常駐管理情報７から削除する。ステップ４７で、動的キャッシュ常駐化機構制御部６は常駐化解除の処理を終了する。

【００６１】ステップ４３で、常駐管理ユティリティプログラム１は、常駐化解除の処理の終了報告を受け取り終了する。

【００６２】

【発明の効果】本発明の常駐管理ユティリティプログラムによれば、キャッシュメモリへのデータセットの常駐／常駐解除指示を、ホストシステムから任意の時点で行うことが可能になる、という効果が得られる。

【００６３】また、常駐するデータの単位として、ボリューム全体、データセット、VTOC領域、物理的な位置で行うことができる、という効果が得られる。

【００６４】また、データセットの削除した場合に、容易にキャッシュメモリ上に残留する領域を解除する手段を提供できる、という効果が得られる。

【図面の簡単な説明】

【図１】本発明の実施の形態である情報処理システムの構成図である。

【図２】本発明の実施の形態である常駐管理ユティリティプログラムによる常駐化設定時の制御の流れを示す図である。

【図３】本発明の実施の形態である常駐管理ユティリティプログラムによる常駐化解除時の制御の流れを示す図である。

【図４】本発明の実施の形態である常駐管理ユティリティプログラムが常駐化設定しているデータセットを削除した場合に常駐化解除を行なう時の制御の流れを示す図である。

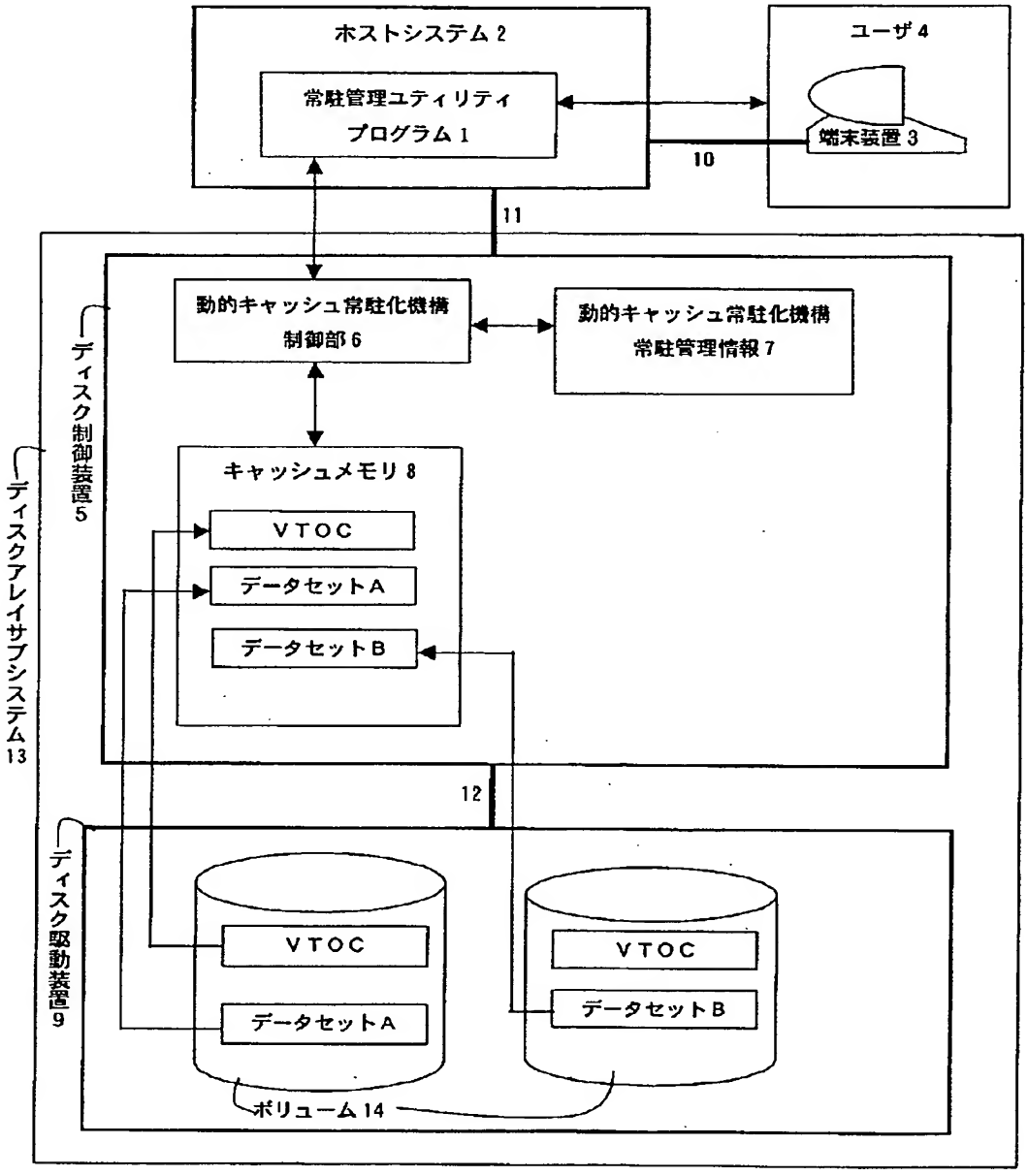
【符号の説明】

１…常駐管理ユティリティプログラム、２…ホストシステム、３…端末装置、４…ユーザ、５…ディスク制御装置、６…動的キャッシュ常駐化機構制御部、７…動的キャッシュ常駐化機構常駐管理情報、８…キャッシュメモリ、９…ディスク駆動装置、10～12…ケーブル、13…ディスクアレイサブシステム、14…ボリューム

【図１】

図１

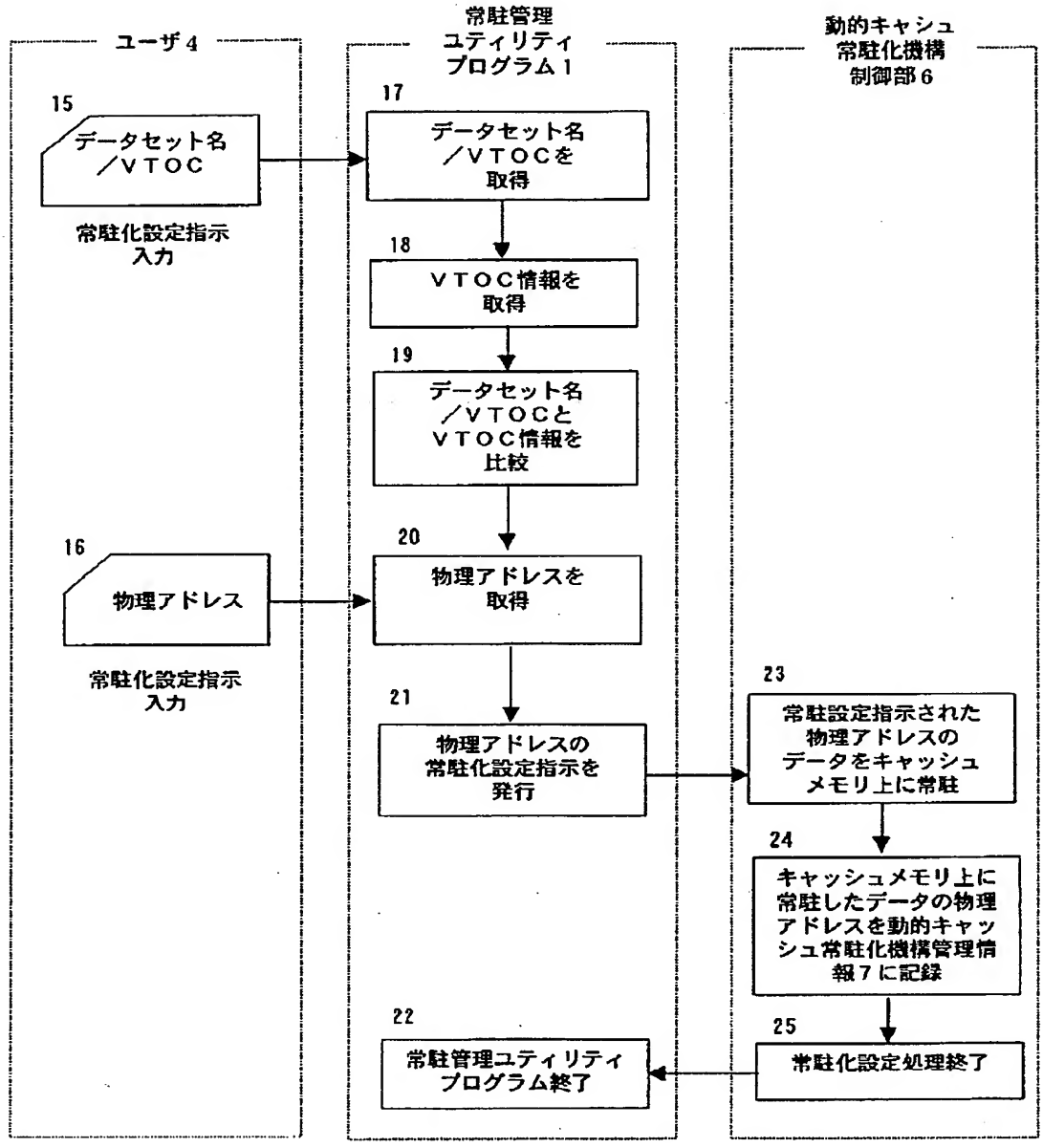
システム構成図



【図２】

図２

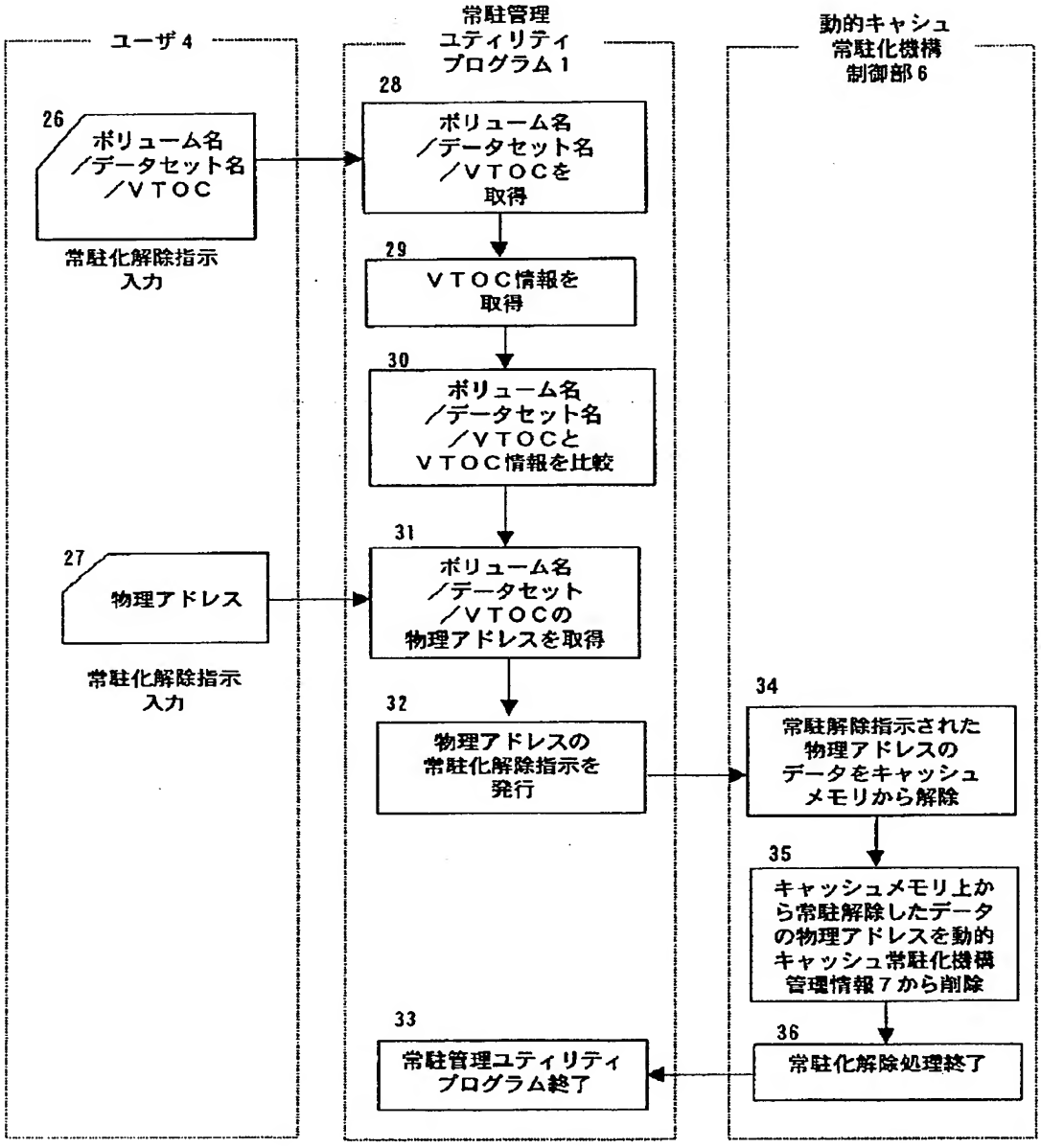
ユティリティプログラムの常駐化設定の制御の流れ



【図３】

図３

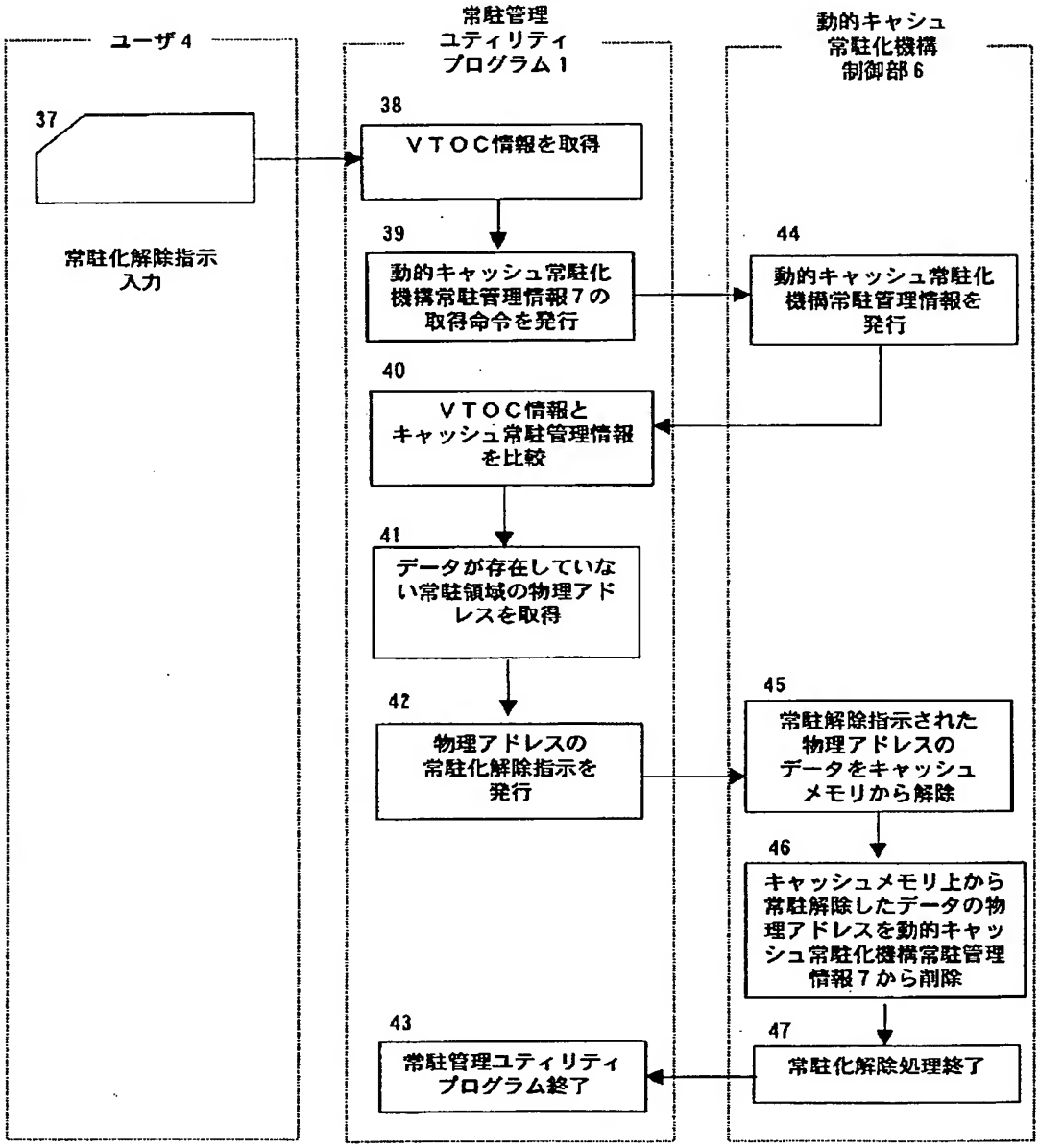
ユティリティプログラムの常駐化解除の制御の流れ



【図４】

図４

ユティリティプログラムのファイル削除後の常駐化解除の制御の流れ




フロントページの続き

| (51)Int.Cl.⁷ | 識別記号 | FI | ﾃｰﾏｺｰﾄﾞ(参考) |
|---|---|---|---|
| Ｇ０６Ｆ　12/12 | ５０３ | Ｇ０６Ｆ　12/12 | ５０３ |
| 13/10 | ３４０ | 13/10 | ３４０Ａ |

Ｆターム(参考) 5B005 JJ11 MM11 QQ05
5B014 EB04 GD13 GD22 GD23
5B065 BA01 CA15 CA30 CC08 CH01
5B082 EA01 EA04 FA12

# United States Patent [19]

**Isobe**

[11] **Patent Number: 4,843,543**

[45] **Date of Patent: Jun. 27, 1989**


[54] **STORAGE CONTROL METHOD AND APPARATUS**


[75] Inventor: **Tadaaki Isobe, Hadano, Japan**

[73] Assignee: **501 Hitachi, Ltd., Tokyo, Japan**



[21] Appl. No.: **40,931**

[22] Filed: **Apr. 21, 1987**

[30] **Foreign Application Priority Data**

Apr. 25, 1986 [JP] Japan ..... 61-96560

[51] **Int. Cl.**[4] ..... **G06F 13/16**

[52] **U.S. Cl.** ..... **364/200**

[58] **Field of Search** ..... 364/200, 900 MS File



[56] **References Cited**

U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 3,626,427 | 12/1971 | MacSorley et al. | 364/200 |
| 3,699,530 | 10/1972 | Capowski et al. | 364/200 |
| 4,325,120 | 4/1982 | Colley et al. | 364/200 |
| 4,661,900 | 4/1981 | Chen et al. | 364/200 |

FOREIGN PATENT DOCUMENTS

| | | |
|---|---|---|
| 60-136849 of 0000 | Japan | GO6F 13/16 |



*Primary Examiner*—Gary V. Harkcom
*Assistant Examiner*—Christopher H. Lynt
*Attorney, Agent, or Firm*—Kenyon & Kenyon



[57] **ABSTRACT**

A storage control device is connected between a number of access request control units and a storage device including a number of memory units. The apparatus includes a number of transmission units, each corresponding to one of the access request control unit. Each transmission unit receives access requests from its associated access request control unit and divides the access requests into a number of groups in the order of issuance from the access request control units. The transmission units also add access request identifiers to the access requests in each group and transmit a number of access requests with access request identifiers to a number of access request deciders. Each access request decider is associated with one of the independently accessible memory units. Each decider receives the access requests directed to its associated memory unit and serially supplies the requests to that memory unit. Additionally, a detection unit is connected to all of the access request deciders in order to detect that all access requests having the same access request identifier, as assigned by the identifier transmission units, have been transmitted from the access request deciders to the corresponding memory units.


**10 Claims, 5 Drawing Sheets**

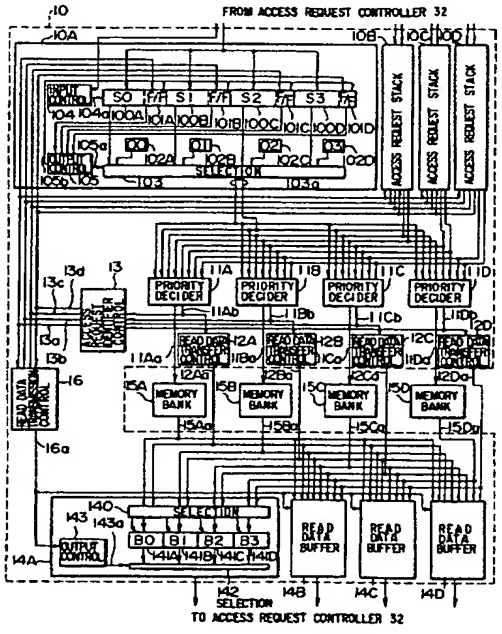


FIG. 1 PRIOR ART

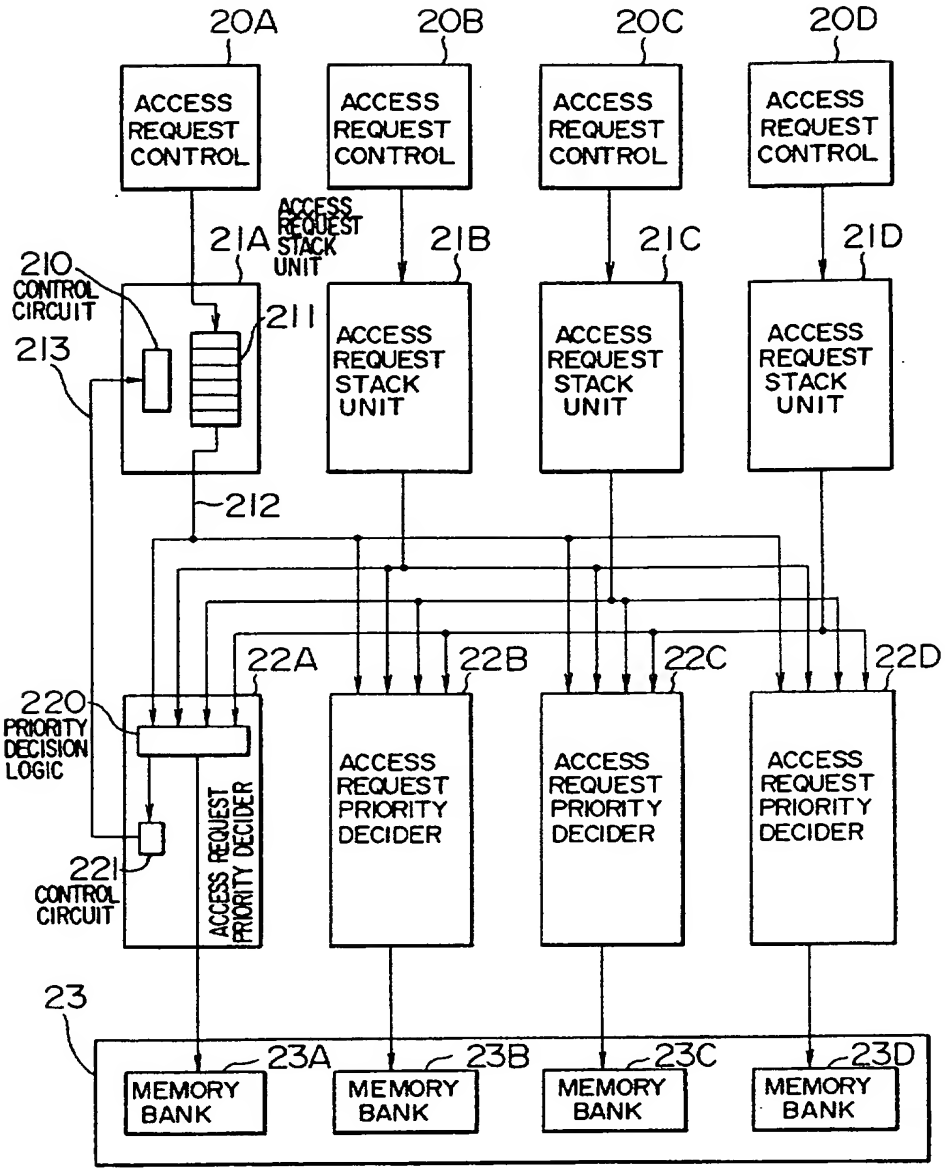


FIG. 2

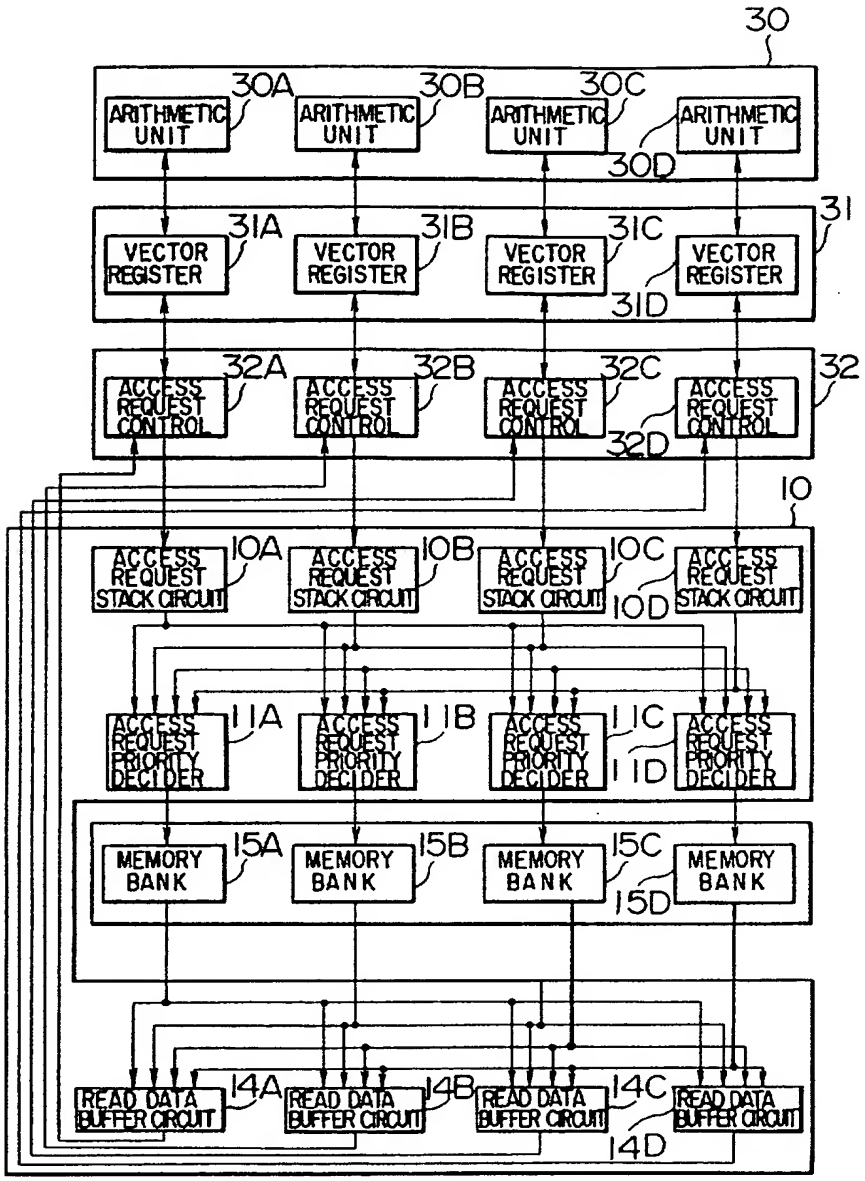

30
30A
30B
30C
30D
ARITHMETIC UNIT
ARITHMETIC UNIT
ARITHMETIC UNIT
ARITHMETIC UNIT
31
31A
31B
31C
31D
VECTOR REGISTER
VECTOR REGISTER
VECTOR REGISTER
VECTOR REGISTER
32
32A
32B
32C
32D
ACCESS REQUEST CONTROL
ACCESS REQUEST CONTROL
ACCESS REQUEST CONTROL
ACCESS REQUEST CONTROL
10
10A
10B
10C
10D
ACCESS REQUEST STACK CIRCUIT
ACCESS REQUEST STACK CIRCUIT
ACCESS REQUEST STACK CIRCUIT
ACCESS REQUEST STACK CIRCUIT
11A
11B
11C
11D
ACCESS REQUEST PRIORITY DECIDER
ACCESS REQUEST PRIORITY DECIDER
ACCESS REQUEST PRIORITY DECIDER
ACCESS REQUEST PRIORITY DECIDER
15A
15B
15C
15D
MEMORY BANK
MEMORY BANK
MEMORY BANK
MEMORY BANK
14A
14B
14C
14D
READ DATA BUFFER CIRCUIT
READ DATA BUFFER CIRCUIT
READ DATA BUFFER CIRCUIT
READ DATA BUFFER CIRCUIT

FIG. 3

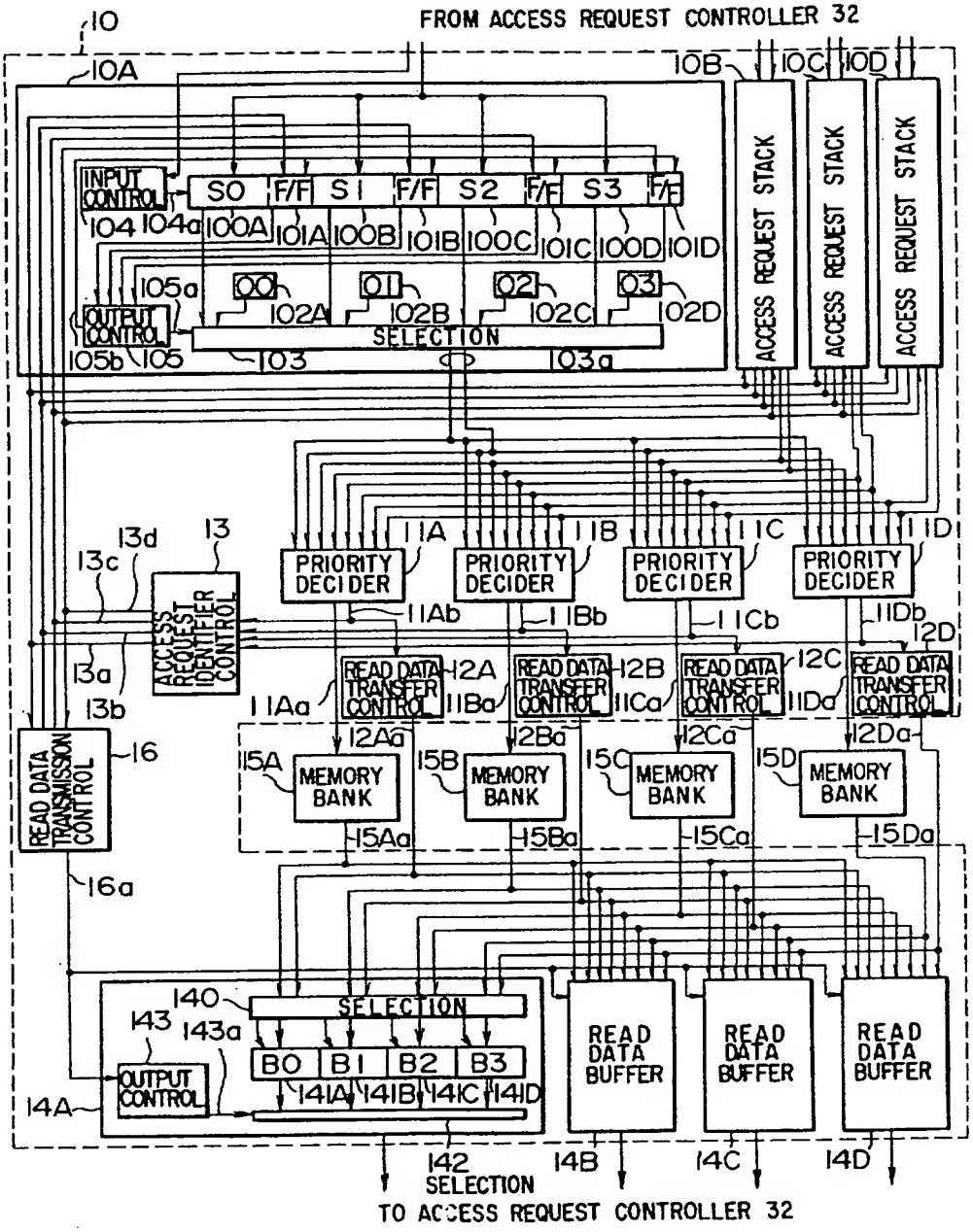

10
FROM ACCESS REQUEST CONTROLLER 32
10A
10B
10C
10D
INPUT CONTROL
104
104a
S0
F/F
S1
F/F
S2
F/F
S3
F/F
100A
101A
100B
101B
100C
101C
100D
101D
ACCESS REQUEST STACK
ACCESS REQUEST STACK
ACCESS REQUEST STACK
105a
00
01
02
03
102A
102B
102C
102D
OUTPUT CONTROL
105b
105
SELECTION
103
103a
13d
13c
13
13a
13b
ACCESS REQUEST IDENTIFIER CONTROL
PRIORITY DECIDER
11A
PRIORITY DECIDER
11B
PRIORITY DECIDER
11C
PRIORITY DECIDER
11D
11Ab
11Bb
11Cb
11Db
12D
READ DATA TRANSFER CONTROL
12A
READ DATA TRANSFER CONTROL
12B
READ DATA TRANSFER CONTROL
12C
READ DATA TRANSFER CONTROL
11Aa
11Ba
11Ca
11Da
12Aa
12Ba
12Ca
12Da
READ DATA TRANSMISSION CONTROL
16
15A
MEMORY BANK
15B
MEMORY BANK
15C
MEMORY BANK
15D
MEMORY BANK
15Aa
15Ba
15Ca
15Da
16a
140
SELECTION
143
143a
B0
B1
B2
B3
OUTPUT CONTROL
14A
141A
141B
141C
141D
READ DATA BUFFER
READ DATA BUFFER
READ DATA BUFFER
142
SELECTION
14B
14C
14D
TO ACCESS REQUEST CONTROLLER 32

FIG. 4

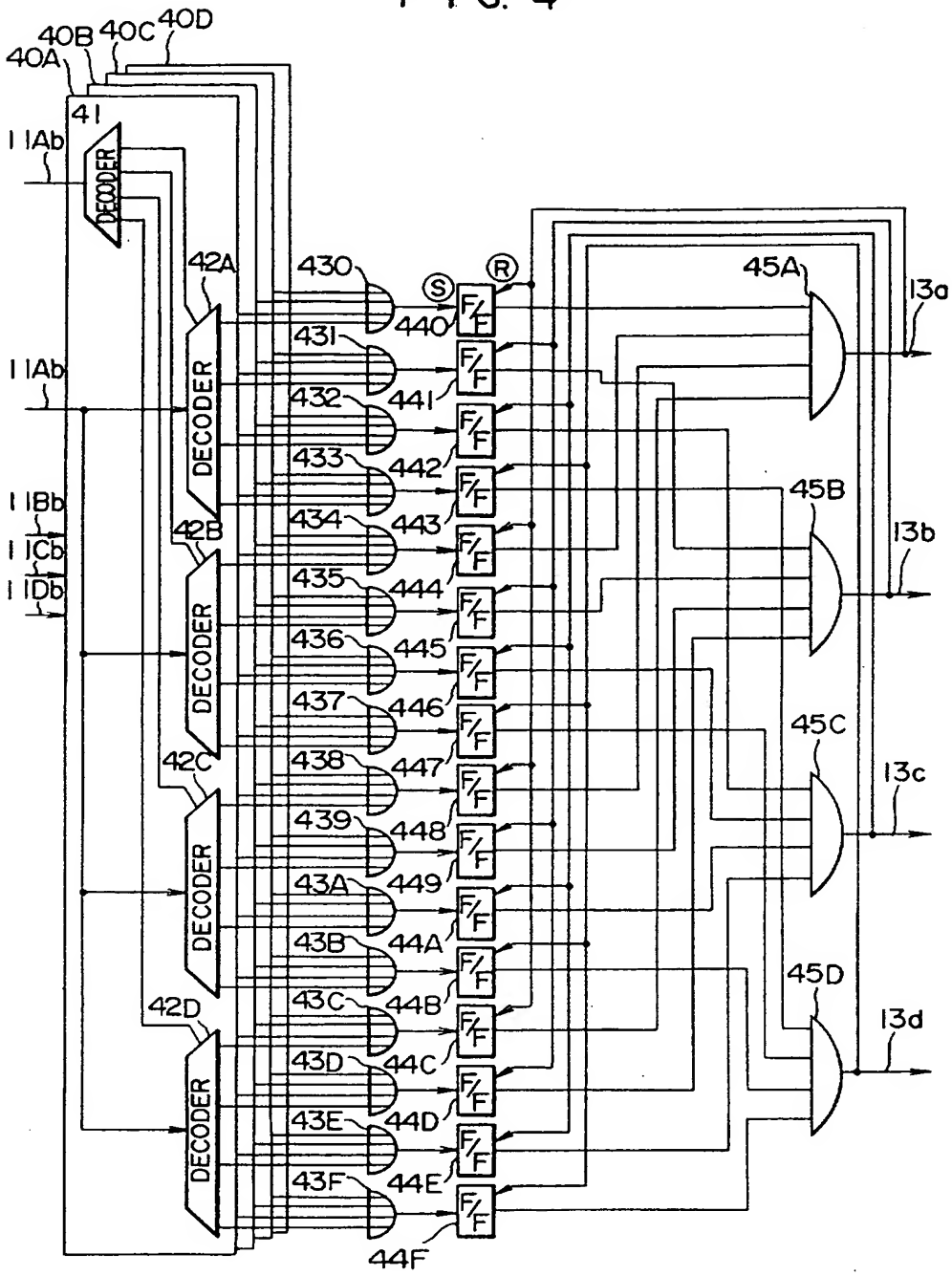
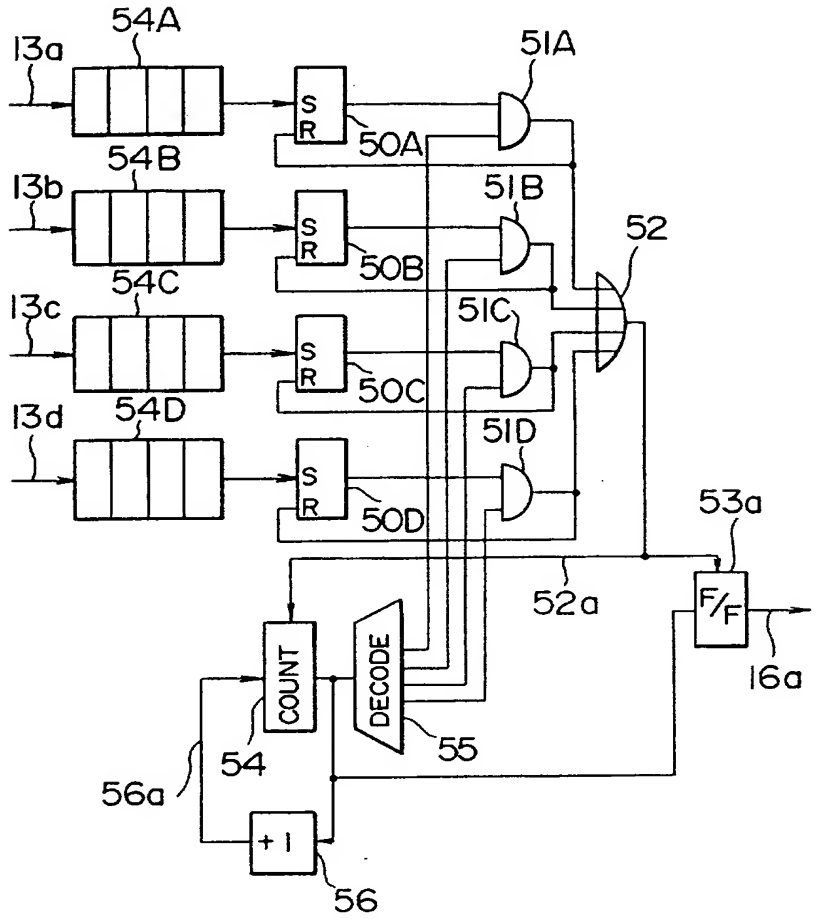

40A
40B
40C
40D
41
11Ab
DECODER
42A
42B
42C
42D
11Bb
11Cb
11Db
430
431
432
433
434
435
436
437
438
439
43A
43B
43C
43D
43E
43F
S
R
F/F
440
441
442
443
444
445
446
447
448
449
44A
44B
44C
44D
44E
44F
45A
45B
45C
45D
13a
13b
13c
13d

FIG. 5


54A
13a
51A
S
R
50A
54B
13b
51B
52
50B
54C
13c
51C
50C
13d
54D
51D
50D
53a
52a
F/F
16a
COUNT
DECODE
54
55
56a
+1
56

# STORAGE CONTROL METHOD AND APPARATUS

## BACKGROUND OF THE INVENTION

This invention relates to storage control method and apparatus for a computer system, especially, being suitable for operating a plurality of access request control units synchronously in parallel and ensuring the sequence or order between access requests which are sequentially issued to a storage area.

In conventional storage controlling, access requests are issued from a plurality of access request control units to a storage comprised of a plurality of independently accessible memory units (memory banks) as will be described below with reference to FIG. 1.

Referring to FIG. 1, access request control units 20A to 20D are sources for issuing access requests. Access request stack units 21A to 21D respectively stack access requests issued from each of the access request control units 20A to 20D. Each of the stack units sends a stacked access request in the order of stacking, to one of access request priority deciders 22A to 22D in accordance with address information contained in that access request. A storage area 23 is comprised of memory banks 23A to 23D.

Taking the access request control unit 20A, for instance, access requests issued from the unit 20A are stacked in a stack circuit 211 of the access request stack unit 21A and under the direction of a control circuit 210, a stacked access request is sent to one of the access request priority deciders 22A to 22D corresponding to one memory bank which is designated by an address contained in that access request. Each access request priority decider 22A, 22B, 22C or 22D selects one of the access requests sent from the access request stack units 21A to 21D and sends a selected access request to the storage area 23. The selection may be accomplished pursuant to priority grading which is predetermined among the access requests. The priority grading may be changed desirably to treat the access requests from the access request stack units as equally as possible. For example, the access request stack units 21A to 21D may be graded in the order of 21A, 21B, 21C and 21D for the initial concurrent access cycle, in the order of 21B, 21C, 21D and 21A for the next concurrent access cycle and in the order of 21C, 21D, 21A and 21B for the concurrent access cycle after next.

In another example, the order of 21A, 21B, 21C and 21D and the order of 21D, 21C, 21B and 21A may be repeated alternately for the purpose of the equal treatment.

It is thus regulated that a plurality of access requests be applied to one decider, for example, 22A at a time and access requests be sent serially one by one from the decider to one memory bank, for example, 23A.

Taking the access request priority decider 22A, for instance, a priority decision logic 220 checks and decides the access requests sent from the access request stack units 21A to 21D to the decider 22A for their priority and transmits a selected one of the access requests to the memory bank 23A of the storage 23. The other access requests not selected at that time are urged to wait at the entrance to the priority decision logic 220.

The access requests are transmitted from the access request control unit 20A until the stack 211 of the access request stack unit 21A fills up. When a control circuit 221 of, for example, the decider 22A transmits a signal



213 indicative of the fact that an access request 212 issued during the preceding machine cycle (a periodical predetermined interval of time during which a group of sequential circuits constituting the system operate synchronously) is selected by the logic 220 and the stack unit 21A receives the signal 213, the succeeding access request 212 is sent from the stack unit 21A. This warrants that in the order of the access requests issued from the access request control unit 20A, accessed data elements can be read out of the storage 23.

In addition to the prior art storage control apparatus described above, another storage control apparatus has been proposed as disclosed in JP-A-No. 60-136849, according to which with a view to improve the performance of the entire system, access requests to be issued from an access request control unit are divided, in the order of issuance, into groups each having a access requests (a is an integer) in unit whereby a access requests in each group are respectively added with access request identifiers 0 to (a−1) and then are issued from the access request control unit, and an access request priority decider directly coupled to a memory unit selects an access request and returns an access request identifier for the selected access request to the access request control unit which is an originator.

Incidentally, a vector processor for fast processing of a scientific computation comprises a plurality of vector registers for holding vector data, a plurality of arithmetic units for operating on the data, and a plurality of access request control units for data transfer between a storage and each of the vector registers, whereby vector elements in one vector instruction are concurrently allocated to a plurality of resources, such as vector registers, arithmetic units or access request control units, and are processed in parallel. Such parallel processing is a so-called element parallel processing mode and has been employed frequently in vector processors.

Generally speaking, it is desirable that the resources in the same group operating concurrently in the element parallel processing mode by completely synchronized with each other to process the allocated elements. By the complete synchronization, a control circuit can be used in common for the plurality of resources in the same group operating in parallel, thereby simplifying the control logic. In this approach, there arises however a problem that a waiting time takes place owing to competition for accessing to memory bank constituting the storage. Accordingly, in order for the plurality of resources in the same group to operate synchronously with each other, the advent of a storage control apparatus is desired which can absorb asynchonism occurring between the resources on account of the waiting time to thereby completely synchronize the storage accessing.

Reviewing then the prior art storage control apparatus disclosed in JP-A-No. 60-136849 mentioned previously, it will be seen that this prior art apparatus presupposes the fact that one access instruction is processed by allocating it to a single access request control unit and fails to take into consideration a processing in which data elements treated by one vector access instruction are divided for allotment to a plurality of access request control units with the aim of being processed in parallel, thus resulting in the problem that the plurality of access request control units to be operated in parallel can not be synchronized with each other so as to process access requests.

## SUMMARY OF THE INVENTION

An object of this invention is to provide a storage controller capable of synchronizing a plurality of access request control units with each other to enable the plural access request control units to issue access requests every clock pitch, in a vector processor or the like in which vector data elements treated by one vector access instruction are divided for allocation to the plurality of access request units and they are then processed.

Another object of this invention is to provide an element parallel process capable of completely suppressing, within a storage controller, asynchronism in access request processing time due to storage accessing which mainly causes disturbance of the synchronous operation for data elements.

Another object of this invention is to provide storage control apparatus and method capable of receiving access requests, issued concurrently from a plurality of access request control units, every clock pitch to greatly improve ability to process the access requests by synchronizing them with each other.

Still another object of this invention is to provide a storage control apparatus and method, based on an element parallel processing mode in which vector elements treated by one vector access instruction and divided for allocation to a plurality of access request control units with the aim of being processed at a time, which can divide access requests issued from the plurality of access request control units into groups each having a access requests in unit and can exchange the order of issuance of the a access requests in unit to a storage while warranting the sequence of read vector data elements.

To accomplish the above objects, according to the invention, access requests issued from a plurality of access request control units are added with access request identifiers, signals are produced when it is detected that access requests with the same access request identifier issued from a plurality of access control units adapted to process one access instruction are all selected by a plurality of access request priority deciders, and the detection signals are used to permit issuance of the succeeding access requests with the same access request identifier, while a plurality of read data elements corresponding to the identifiers are transmitted concurrently to the access request originators after expiration of a fixed time required for the detection signals to access a storage.

Especially where the access operation to a series of elements of vector-like data is processed by dividing the vector data elements for allocation to a plurality of access request control units, access requests to be issued from respective access request control units are divided, in the order of request issuance, into groups each having a access requests in unit and the a access requests in each group are respectively added with access request identifiers 0 to (a−1), while a plurality of access request priority deciders respectively coupled to memory units select all access requests with the same access request identifer issued from the plurality of access request control units adapted to division process one access instruction and return the access identifiers for the selected access requests to the plurality of access request control units which are originators. This ensures that each of the plurality of access request control units for division processing one instruction can perform sequential issuance of at least a access requests regardless of whether the a access requests are selected by the access request priority deciders. Further, when access requests (with identifiers "0") issued, during the first issue cycle, from the plurality of access request control units adapted to division process one instruction are all selected by the deciders and the signals to this effect are then returned to the access request control units, the access request control units are allowed to issue access requests with identifiers "0" for the second issue cycle following issuance of access requests with the previous identifiers "a−1" for the first issue cycle. The succeeding issue cycles are taken care of in a similar way so that the access request control units for division processing one instruction can be synchronized with each other to sequentially issue access requests.

In a read data controller for transmitting read data elements corresponding to access requests to the corresponding access request originators, identifiers attendant on read data elements are recognized and data elements are stored in buffer positions corresponding to the identifiers. When a fixed time (access time of the storage) expires following detecting that all of the access requests with the same identifier issued from the plurality of access request control units adapted to division process one instruction are transmitted to the storage, that is, at the time that data elements of the same identifier are all stored in the read data buffers, the data elements stored in the buffers are transmitted synchronously in parallel, in the order to identifiers, to the data originators. This ensures that a plurality of read data elements corresponding to access requests issued from the plurality of access request control units adapted to division process one instruction can be transmitted while warranting the sequence of the read data elements.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram showing a prior art storage control apparatus.

FIG. 2 is a block diagram showing the essential part of a computer system employing a storage controller according to the invention.

FIG. 3 is a circuit diagram showing a storage controller according to an embodiment of the invention.

FIG. 4 is a circuit diagram showing an embodiment of an access request identifier control circuit shown in FIG. 3.

FIG. 5 is a circuit diagram showing an embodiment of a read data transmission control circuit shown in FIG. 3.

## DESCRIPTION OF THE PREFERRED EMBODIMENTS

The invention will now be described by way of example with reference to the accompanying drawings.

FIG. 2 illustrates, in block form, an exemplary construction of the essential part of a computer system incorporating an embodiment of the invention. The computer system comprises an arithmetic section **30** including a plurality of (four in this embodiment) arithmetic units **30A** to **30D**, a vector register section **31** including vector registers **31A** to **31D** and adapted to serve as a data buffer between the arithmetic section **30** and a storage **15**, an access request controller **32** including access request control units **32A** to **32D**, and a storage controller **10**. The storage **15** includes a plurality of (four in this embodiment) independently accessible memory banks **15A** to **15D**. Address information atten-

dant on an access request is decoded to determine which memory bank should be accessed. The storage controller 10 includes access request stack circuits 10A to 10D in association with the access request control units 32A to 32D, read data buffer circuits 14A to 14D, and access request priority deciders 11A to 11D in association with the memory banks 15A to 15D.

The operation of the computer system shown in FIG. 2 will be described below by referring to, as an example, a storage read operation based on the element parallel processing mode.

Firstly, when reading a piece of vector data from the storage and storing the vector data in the vector register unit 31, vector elements are allocated to the access request control units 32A to 32D in the following manner and access requests are generated.

Access request

Control Units

32A: 0th, 4th, 8th, . . . 4nth elements
32B: 1st, 5th, 9th, . . . $(4n+1)$th elements
32C: 2nd, 6th, 10th, . . . $(4n+2)$th elements
32D: 3rd, 7th, 11th, . . . $(4n+3)$th elements (n being positive integer)

Access requests concurrently generated for four elements (for example, the 0th, 1st, 2nd and 3rd elements) are respectively sent to the corresponding access request stack circuits 10A to 10D. In accordance with an address contained in an access request, each stack circuit sends that access request to one of the priority deciders 11A to 11D. If a plurality of access requests compete with one another in either of the priority deciders, that priority decider selects one access request pursuant to predetermined priority grading and then sends the selected access request to corresponding one of the memory banks 15A to 15D. The read data corresponding to the access requests sent to the memory banks are read and transferred to the storage controller 10 after expiration of a fixed time (corresponding to an access time), so that elements of the read data representative of the vector elements are set to the read data buffer circuits 14A to 14D corresponding to the access request control circuits 32A to 32D. When all the data elements corresponding to the four access requests concurrently issued from the access request control circuits 32A to 32D are read out of the storage 15, these read data elements are sent to the access request control units. This data group is sent in the order of the concurrent issuance and then stored concurrently in the vector registers 31A and 31D. Allocation of the vector elements to the vector register unit 31 is as follows.

Vector registers
31A: 0th, 4th, 8th, . . . 4nth elements
31B: 1st, 5th, 9th, . . . $(4n+1)$th elements
31C: 2nd, 6th, 10th, . . . $(4n+2)$th elements
31D: 3rd, 7th, 11th, . . . $(4n+3)$th elements

For arithmetic operation of the data stored in the vector registers 31A to 31D, the vector elements are allocated to the arithmetic units 30A to 30D in the following manner and operation results are again stored in the vector registers.

Arithmetic units
30A: 0th, 4th, 8th, . . . 4nth elements
30B: 1st, 5th, 9th, . . . $(4n+1)$th elements
30C: 2nd, 6th, 10th, . . . $(4n+2)$th elements
30D: 3rd, 7th, 11th, . . . $(4n+3)$th elements

In the above arithmetic operation, the four arithmetic units 30A to 30D are completely synchronized with each other for operation. For example, results for the 0th, 1st, 2nd and 3rd elements are obtained concurrently and stored in the vector registers 31A to 31D.

When writing the data stored in the vector registers 31A to 31D into the storage 15, the vector elements are allocated to the access request control units 32A to 32D as in the case of data read previously described and four elements, for example, the 0th, 1st, 2nd and 3rd elements are sent to the corresponding access request stack circuits 10A to 10D. Thereafter, access requests are sent to the storage 15 similarly to the read operation.

As described above, the four arithmetic units 30A to 30D are synchronized with each other, the four vector registers 31A to 31D are synchronized with each other and the four access request control units 32A to 32D are synchronized with each other, so as to process the vector elements. Accordingly, the element parallel processing mode based on synchronous operation is permitted to take a logical construction which uses a single control system logic to control a set of the four arithmetic units 30A to 30D, a set of the four vector registers 31A to 31D and a set of the four access request control units 32A to 32D.

Within the storage controller 10, however, the four access requests concurrently issued from the access request control units 32A to 32D operating in synchronism are not always processed concurrently because of a condition of an accessed memory bank (for example, the memory bank being occupied by the preceding access request) and competition with another access and as a result, the access results are sent to the memory banks at different times. Accordingly, a control system is needed wherein the read data buffer circuits 14A to 14D of the storage controller 10 wait until the read data elements corresponding to the access requests concurrently sent from the access request control units 32A to 32D are all stored in the buffers 14A to 14D and when all the data elements are stored, the data elements are sent to the four read data access request control units 32A to 32D at a time.

A synchronous control mode in the storage controller will now be described in greater detail with reference to FIG. 3.

The construction of the storage controller 10 of FIG. 2 is exemplarily detailed in FIG. 3, along with connection to the memory banks 15A to 15D. In addition to the access request stack circuits 10A to 10D, priority deciders 11A to 11D and read data buffer circuits 14A to 14D, the storage controller 10 comprises read data transfer control circuits 12A to 12D and an access request identifier control circuit 13.

Four access requests for, for example, the 0th, 1st, 2nd and 3rd elements issued concurrently from the access request control units 32A to 32D arrive at the access request stack circuits 10A to 10D, respectively. The access request reaching the access request stack circuit 10A, for example, is set in one of stacks S0 (100A), S2 (100B), S2 (100C) and S3 (100D), for example, the stack S0 (100A) which is designated by an input control circuit 104. The input control circuit 104 designates one of the stacks by sending thereto a signal 104*a* indicative of one of stack positions "0" to "3" (corresponding to the stacks S0 to S3) to which one access request is to be stored. The indication of the signal 104*a* changes to circulate through "0"→"1"→"2"→"3"→"0" . . . each time that one access request is stored in one stack, thereby designating a subsequent store stack position.

Each access request stored in one of the stacks S0 to S3 of each of the access request stack circuits 10A to 10D is sent to the priority deciders 11A to 11D through a selection circuit 103 under the control of an output control circuit 105. For example, when the output control circuit 105 designates the stack position or number "0" representative of the stack S0 (100A), an access request stored in the stack S0 (100A) is delivered. Like the input control circuit 104, the output control circuit 105 designates one of the stacks by sending to the selection circuit 103 a signal 105*a* indicative of one of stack positions "0" to "3" from which one access request is to be delivered. The indication of the signal 105*a* changes to circulate through "0"→"1"→"2"→"3"→"0" . . . each time that one access request is delivered to the selection circuit 103. However, the output control circuit 105 is different from the input control circuit 104 in that it controls the delivery of the access request from the designated stack dependent on contents of access request transmission control flip-flops 101A to 101D associated with the stacks S0 to S3, respectively.

More particularly, when the output control circuit 105 accesses, for example, the access request transmission control flip-flop 101A associated with the stack S0 (100A) and the content of the flip-flop 101A is "1", an access request stored in the stack S0 (100A) is delivered to the priority deciders 11A to 11D. Contrarily, when the content of the flip-flop 11A is "0", the delivery of the access request from the stack S0 (100A) is inhibited until the flip-flop 101A is set to "1" by a signal 13*a* from the access request identifier control circuit 13 and at the same time, the stack number "0" indicated by the signal 105*a* delivered out of the output control circuit 105 is maintained.

Where the content of the flip-flop 101A is "1", after the access request stored in the assodiated stack S0 (100A) has been fetched and delivered to the priority deciders 11A to 11D, the output control circuit 105 resets the flip-flop 101A to "0" by sending thereto a signal 105*b* representative of information about completion of that delivery. Then, the output control circuit 105 is ready to access the subsequent stack S1.

Taking the access request stack circuit 10A, for instance, an access request 103*a* sent from the selection circuit 103 to the priority deciders 11A to 11D contains, in addition to address information stored in for example the stack S0 (100A), one of access request identifiers of two digits "00", "01", "02" and "03" delivered out of access request identifier generators 102A to 102D, the first digit "0" of the access request identifier being indicative of the access request stack circuit 10A in question and the second digit "0", "1", "2" or "3" being the stack number designating the stack S0, S1, S2 or S3. In this example, the access request identifier "00" is designated.

Although the access request stack circuit 10A has been described as an example, each of the remaining stack circuits 10B to 10D operates similarly to the stack circuit 10A. Thus, four access requests issued concurrently from the four access request control units are stored and processed in the corresponding stacks of the same number of respective stack circuits.

The access request 103*a* sent from the access request stack circuit 10A can reach one of the priority deciders 11A to 11D in association with the memory banks, for example, the decider 11A in accordance with an access address designated by that access request. The access request 103*a* reaching the priority decider 11A is checked for its priority in relation to another access request sent from the remaining access request stack circuit 10B, 10C or 10D and if selected, an access request 11A*a* is sent to the corresponding memory bank 15A, together with information indicative of either read or write mode, an address on the memory and data in the case of write mode.

Concurrently with the delivery of the access request 11A*a*, an access request identifier 11A*b* is sent to the access request identifier control circuit 13. Details of the access request identifier control circuit 13 are illustrated in FIG. 4. Referring to FIG. 4, the access request identifier 11A*b* is supplied to a decoder circuit 40A which includes a decoder 41 for decoding a number representative of the access request stack circuit 10A, 10B, 10C or 10D and decoders 42A to 42D for decoding a stack number in each circuit 10A, 10B, 10C or 10D and which is operative to specify the access request stack circuit and the stack number. Signals from the decoder circuit 40A are passed through OR gates 430 to 43F so as to set to "1" one of flip-flops 440 to 44F which are adapted to indicate validity of access request identifier for each access request stack circuit specified by the decoder circuit. Similarly, access request identifiers 11B*b* to 11D*b* are decoded by decoder circuits 40B to 40D and used to set specified flip-flop for indication of validity of access request identifier.

When the flip-flop corresponding to the stacks of the same number in respective access request stack circuits 10A to 10D, for example, the flip-flops 440, 444, 448 and 44C are all rendered "1", indicating that the four access requests reaching the access request stack circuits 10A to 10D at a time and set in the stacks S0 in respective stack circuits are all sent to the storage 15, all of the "1" signals are detected by an AND gate 45A (similar AND gates 45B to 45D are provided) to produce the signal 13*a* which sets to "1" the contents of the access request transmission control flip-flops 101A associated with the stacks S0 in respective access request stack circuits 10A to 10D. This ensured that the subsequent access requests stored in the stacks S0 in respective access request stack circuits 10A to 10D are ready for transmission to the priority deciders 11A to 11D. At the time the signal 13*a* is sent to the access request stack circuits 10A to 10D, the validity indication flip-flops 440, 444, 448 and 44C producing the signal 13*a* are reset to "0".

Incidentally, read data elements 15A*a* to 15D*a* corresponding to the access requests 11A*a* to 11D*a* sent to the memory banks 15A to 15D are transferred to the read data buffer circuits 14A to 14D, together with access request identifiers 12A*a* to 12D*a* retarded at the read data transfer control circuits 12A to 12D by an access time of the memory banks and the stack numbers. On the basis of a stack number and an access request stack circuit number designated by the access request identifier, each of the read data buffer circuits 14A to 14D stores, under the direction of a selection circuit 140, one of the read data elements into one of data buffers B0 (141A) to B3 (141D) of each of the read data buffer circuits 14A to 14D respectively corresponding to the access request stack circuits 10A to 10D, the data buffers being provided in correspondence to the stacks S0 to S3 of each of the access request stack circuits 10A to 10D. For example, the identifier 12A*a* indicative of "00" serves to store the read data element 15A*a* in the data buffer B0 (141A) of the read data buffer circuit 14A. In the event that the number of access requests designated by an access instruction is too small to oc-

cupy all of the memory banks, some of the access requests 11A*a* to 11D*a* sent to unoccupied memory banks are added with non-operation codes to thereby complete the access instruction.

Incidentally, a read data transmission control circuit **16** operates independently of the storage of the read data elements to the data buffer circuits to concurrently transmit the four read data elements sorted in the read data buffer circuits **14A** to **14D** to the access request control circuits **32A** to **32D**. FIG. **5** illustrates details of the read data transmission control circuit **16**.

Sent to the read data transmission control circuit **16** are the signals **13***a* to **13***d* from the access request identifier control circuit **13** respectively indicating that access requests stored in the stacks of the same number in respective four access request stack circuits **10A** to **10D** are all transferred to the memory banks. Each of the signals **13***a* to **13***d* indicates, in other words, that read data elements are all stored in corresponding one of the four data buffers **B0** (**141A**) to **B3** (**141D**) in respective four read data buffer circuits **14A** to **14D** after expiration of a fixed time (corresponding to the access time of each memory bank). Accordingly, validity of the read data elements in corresponding one of the data buffers **B0** (**141A**) to **B3** (**141D**) in respective read data buffer circuits **14A** to **14D** can be indicated by passing the signals **13***a* to **13***d* through buffers **54A** to **54D** which delay these signals to match with the memory bank accessing so as to set flip-flop **50A** to **50D**.

The read data transmission control circuit **16** is operative to sent to the read data buffer circuits **14A** to **14D** a control signal **16***a* which controls sequential delivery of the valid data elements from the read data buffers, beginning with the data buffer **B0** (**141A**) followed by the data buffers **B1** (**141B**), **B2** (**141C**) and **B3** (**141D**). But when the content of a flip-flop for indication of read data validity **50A**, **50B**, **50C** or **50D** corresponding to a data buffer designated by the content of read data pickup pointer **54** which is decoded by a decoder **55** is "0", that is, when a read data element corresponding to at least one of four access requests which is stored in a stack within the access request stack circuits **10A** to **10D**, the position of that stack being designated by the pointer **54**, has not reached the read data buffer yet, the delivery of the read control signal **16***a* is inhibited by AND gates **51A** to **51D** and the content of the pointer **54** also remains unchanged until that flip-flops **50A**, **50B**, **50C** and **50D** are set to "1". Subsequently, when the content of the flip-flop for indication of read data validity corresponding to the data buffer designated by the pointer **54** changes to "1", this change is detected by the AND gates **51A** to **51D** and an OR gate **52**, thereby, enabling the read control signal **16***a* to be transmitted through a flipflop **53A**. Concurrently therewith, a signal **52***a* resets the read data validity indication flip-flop designated by the pointer **54** to "0" and the content of the pointer **54** is renewed by being added with "+1" from a +1 circuit **56**.

The four read data buffer circuits **14A** to **14D** receive the read control signal **16***a* simultaneously so that the read control signal **16***a* is applied to an output control circuit **143** provided in each circuit **14A**, **14B**, **14C** or **14D**. The output control circuit **143** then issues a signal **143***a* which controls a selection circuit **142** such that a read data element stored in a data buffer whose number or position is designated by the read control signal **16***a* can be picked up.

Through the operation described above, four read data elements can be transmitted synchronously in parallel from the four read data buffer circuits **14A** to **14D** to the access request control units **32A** to **32D**.

As has been described, according to the present embodiment, of four access requests issued concurrently and synchronously from the four access request control units **32A** to **32D**, a maximum of three access requests can be overridden within the storage controller **10** so that the four access requests can be processed every clock pitch, and besides read data elements corresponding to the four access requests received at a time can be sent to the access request control units **32A** to **32D** in the order of the synchronous, parallel receipt of the access requests.

I claim:

**1**. A storage control apparatus, connected between a storage device, said storage device including a plurality of independently accessible memory units, and a plurality of access request control units, each of said plurality of access request control units being operative to issue, to said storage device, access requests corresponding to elements of data, said storage control apparatus sequentially transmitting selected ones of said access requests issued from said plurality of access request control units to corresponding memory units designated by said access requests, said storage control apparatus comprising:

a plurality of identifier transmission means for dividing said access requests issued from said plurality of access request control units into a plurality of groups based on the order of issuance from said access request control units wherein each identifier transmission means is associated with one of said access request units and divides said access requests from said associated access request unit, and wherein each identifier transmission means further adds access requests in each group with access request identifiers, and wherein each identifier transmission means further transmits a plurality of access requests with access request identifiers added thereto;

a plurality of access request decider means, each access request decider means being provided in correspondence to one of said plurality of memory units, for receiving the plurality of access requests with access request identifiers sent from said identifier transmission means and serially supplying to the corresponding memory units the received access requests one by one; and

means, connected to said plurality of access request decider means, for detecting that all access requests with the same access request identifier have been transmitted from said plurality of access request decider means to said corresponding memory units, said means for detecting further sending to all of said identifier transmission means information indicative of the detection that all access requests with the same access request identifier have been transmitted from said plurality of access request decider means to said corresponding memory units which permits said same access request identifier to be used for new access requests.

**2**. The storage control apparatus of claim **1** further comprising:

a plurality of buffer means, each one of said buffer means being associated with one of said plurality of

memory units, for holding respective data elements read out of respective memory units; and

means, receiving from said plurality of identifier transmission means signals indicative of the fact that all of the access requests having the same access request identifier have been detected, for informing said plurality of buffer means of said detection indicative signals so as to permit said read data elements held in said plurality of buffer means to be transmitted to said plurality of access request control units.

3. The storage control apparatus of claim 2 wherein said each identifier transmission means comprises a stack for access requests for said data elements received from said associated access request control unit, and a plurality of flip-flops associated with each stack for controlling whether the access requests held in said stacks are permitted to be transmitted to said access request decider means, and wherein said plurality of buffer means each have a plurality of buffer positions for holding said data elements, and wherein each said stack is partitioned into a number of portions wherein said number of portions correspond to the number of access request identifiers, each portion corresponding to one of said identifiers, each one of said plurality of flip-flops being associated with one of said portions of said stacks and each buffer position being associated with one of said access request identifiers.

4. The storage control apparatus of claim 1 wherein each said access reques decider means is coupled to all of said plurality of identifier transmission means and wherein each said access request decider means comprises priority means for serially delivering a plurality of access requests that are competing with each other to said memory units in accordance with a predetermined priority value.

5. The storage control apparatus of claim 4 wherein said priority means comprises means for sequentially changing the priority value at the rate of a predetermined number of data access operations.

6. The storage control apparatus of claim 2 further comprising means for sending, from respective access request decider means to respective buffer means, an access request identifier and an identification number, which identifier number identifies the identifier transmission means responsible for a given access request, in synchronization with a data element to be read out of the memory unit associated with said access request decider means wherein said data element is read out in response to said given access request.

7. The storage control apparatus of claim 6 wherein said means for informing comprises means for providing a plurality of signals, one associated with each of said buffer means, with a delay time corresponding to an interval of time within which data elements have been read out of respective memory units and then transferred to respective buffer means, said plurality of signals indicating that all access requests having the same request identifier have been sent to said memory units.

8. The storage control apparatus of claim 1 wherein said detecting means comprises,

a plurality of latch means for latching a flag indicative of the validity of a detected identifier, the number of said latch means being equal to the product of the number of said identifier transmission means and the number of access request identifiers in each identifier transmission means,

decoders connected to said plurality of latch means, for discriminating the identification of said identifier transmission means based on said identification number, and

decoders for discriminating individuality of said access request identifiers.

9. A computer system comprising:

a storage device having a plurality of independently accessible memory units;

a plurality of access request control units operative to issue, to said storage device, access requests corresponding to elements of vector data;

a plurality of identifier transmission means, each identifier transmission means being associated with one of said plurality of access request control units, for dividing a plurality of access requests received from said access request control units into a plurality of groups based on the order of reception, wherein each said identifier transmission means transmits the plurality of access requests together with access request identifiers for identifying individuality of said access requests;

a plurality of access request decider means each associated with one of said memory units, for selectively receiving said access requests in accordance with addresses in said storage device and serially supplying to the corresponding one of said memory units the selected access requests one by one; and

means, connected to the output of all of the plurality of access request decider means, for detecting that access requests with the same access request identifier have all been transmitted from said plurality of access request decider means to said address corresponding memory units in said storage.

10. A storage control method for controlling an access sequence to respective elements of vector data in an orderly manner in a computer system including a storage device, having a plurality of independently accessible memory units, and a plurality of access request issue units for issuing access requests to said storage device, said access requests corresponding to the vector data elements, said method comprising the steps of:

dividing the access requests corresponding to the elements into a plurality of groups based on the order of request issuance;

adding access requests in each group with access request identifiers which are common to respective groups;

receiving access requests with the same identifier sent to respective memory units and deciding whether all of the access requests with the same identifier have been sent to respective memory units; and

allowing, when the access requests with the same identifier have been sent to respective memory units the use of the same identifier for new vector data elements.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

# CERTIFICATE OF CORRECTION

PATENT NO. : 4,843,543



DATED : June 27, 1989

INVENTOR(S) : Isobe

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 1, line 14 change "storage " to --storage area--; line 30 change "th." to --the--; line 62 change "storage 23." to --storage area 23.--.
Column 2, line 10 change "storage 23" to --storage area 23--;
line 41 change "by" to --be--.
Column 4, lines 64 and 66 change "storage 15" to --storage area 15--.
Column 5, line 30 change ".equest" to --request--.
Column 6, lines 5 and 11 change "storage 15" to --storage area 15--;
line 58 change "S2" (first occurrence) to --$S_1$--.
Column 7, line 30 change "\" to --is--;
line 36 change "assodiated" to --associated--.
Column 9, line 30 change "flip-flop" to --flip-flops--;
line 32 change "sent" to --send--;
line 48 change "that" to --the--.
Column 10, line 67 change "of.buffer" to --of buffer--.
Column 11, line 30 change "reques." to --request--.

# UNITED STATES PATENT AND TRADEMARK OFFICE
# CERTIFICATE OF CORRECTION



PATENT NO. : 4,843,543

DATED : June 27, 1989

INVENTOR(S) : Isobe

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 12, line 39 change "sponding memory" to --sponding to memory--.

**Signed and Sealed this**

**Twenty-sixth Day of June, 1990**

*Attest:*

*Attesting Officer*

HARRY F. MANBECK, JR.

*Commissioner of Patents and Trademarks*

US006529944B1

(12) **United States Patent**
**LeCrone**

(10) **Patent No.: US 6,529,944 B1**
(45) **Date of Patent: Mar. 4, 2003**

(54) **HOST SYSTEM FOR REMOTE CONTROL OF MASS STORAGE VOLUMES USING CASCADING COMMANDS**

(75) Inventor: **Douglas LeCrone**, Hopkinton, MA (US)

(73) Assignee: **EMC Corporation**, Hopkinton, MA (US)

( * ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/225,461**

(22) Filed: **Jan. 6, 1999**

(51) **Int. Cl.**[7] ..... **G06F 15/16**
(52) **U.S. Cl.** ..... **709/211**; 707/10; 707/202; 707/204; 707/205; 711/159; 711/162
(58) **Field of Search** ..... 709/211; 707/10, 707/204, 202, 205; 711/162, 159

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 4,156,907 A | * | 5/1979 | Rawlings et al. | 709/212 |
| 4,688,171 A | * | 8/1987 | Selim et al. | 709/208 |
| 4,747,047 A | * | 5/1988 | Coogan et al. | 710/38 |
| 5,566,331 A | * | 10/1996 | Irwin, Jr. et al. | 707/10 |
| 5,852,718 A | * | 12/1998 | Loo | 709/208 |
| 6,000,825 A | * | 12/1999 | Fredriksson | 364/138 |
| 6,101,497 A | * | 8/2000 | Ofek | 707/10 |
| 6,112,229 A | * | 8/2000 | Hardy et al. | 709/206 |
| 6,148,382 A | * | 11/2000 | Bitner et al. | 711/162 |
| 6,161,111 A | * | 12/2000 | Mutalik et al. | 707/205 |
| 6,202,135 B1 | * | 3/2001 | Kedem et al. | 711/162 |

* cited by examiner

*Primary Examiner*—Ayaz Sheikh
*Assistant Examiner*—Frantz B. Jean
(74) *Attorney, Agent, or Firm*—John M. Gunther; Penelope S. Wilson

(57) **ABSTRACT**

A host system for remote control of mass storage volumes using cascading commands which collect information about linked remote volumes located in physically separate sites so other cascading commands can be issued to effect changes in linked remote volumes. A host computer program issues the cascading commands which ask the locally communicating mass storage system to return information which can be used to identify one or more levels of remote mass storage systems in a stream of remote mass storage systems. Once a mass storage system at a given level has been identified, commands can be sent by the host through the locally communicating mass storage system to cause actions to occur at the identified remote level, whether or not there are multiple intervening levels of remote mass storage systems. In one embodiment, a host computer can query, establish, split, re-establish, copy, restore, reverse split, differentially split and reverse differentially split business continuance volumes at any level in a stream of local and remote mass storage system sites.

**20 Claims, 17 Drawing Sheets**

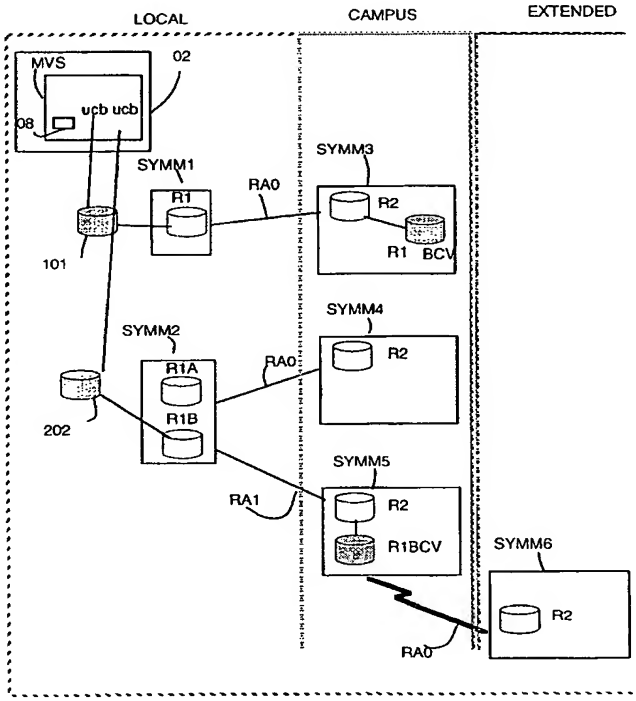


*Fig. 1*

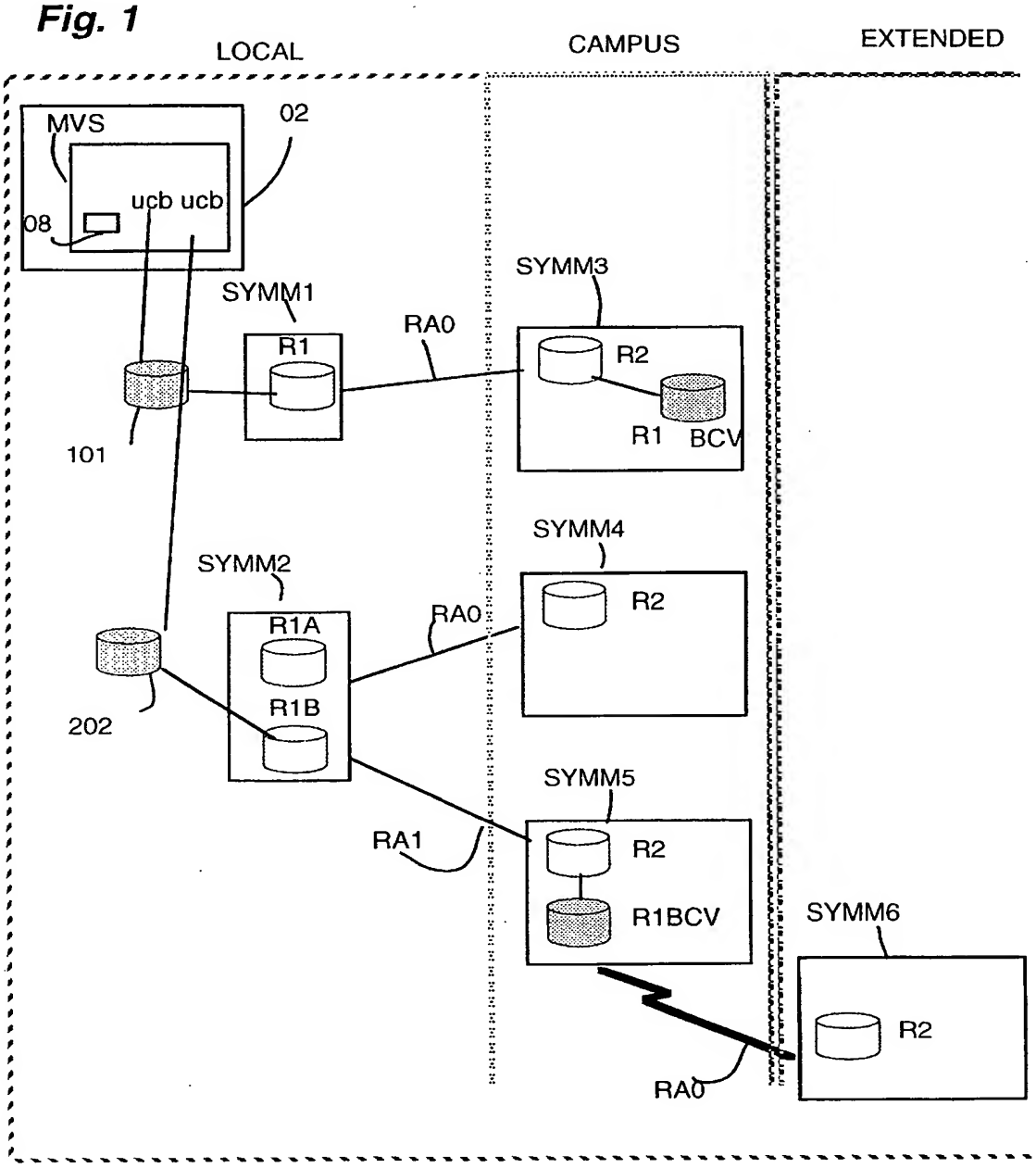

LOCAL
CAMPUS
EXTENDED
MVS
02
ucb ucb
08
SYMM1
R1
RA0
SYMM3
R2
R1
BCV
101
SYMM4
SYMM2
R2
R1A
RA0
R1B
202
SYMM5
RA1
R2
R1BCV
SYMM6
R2
RA0

*Fig. 2*

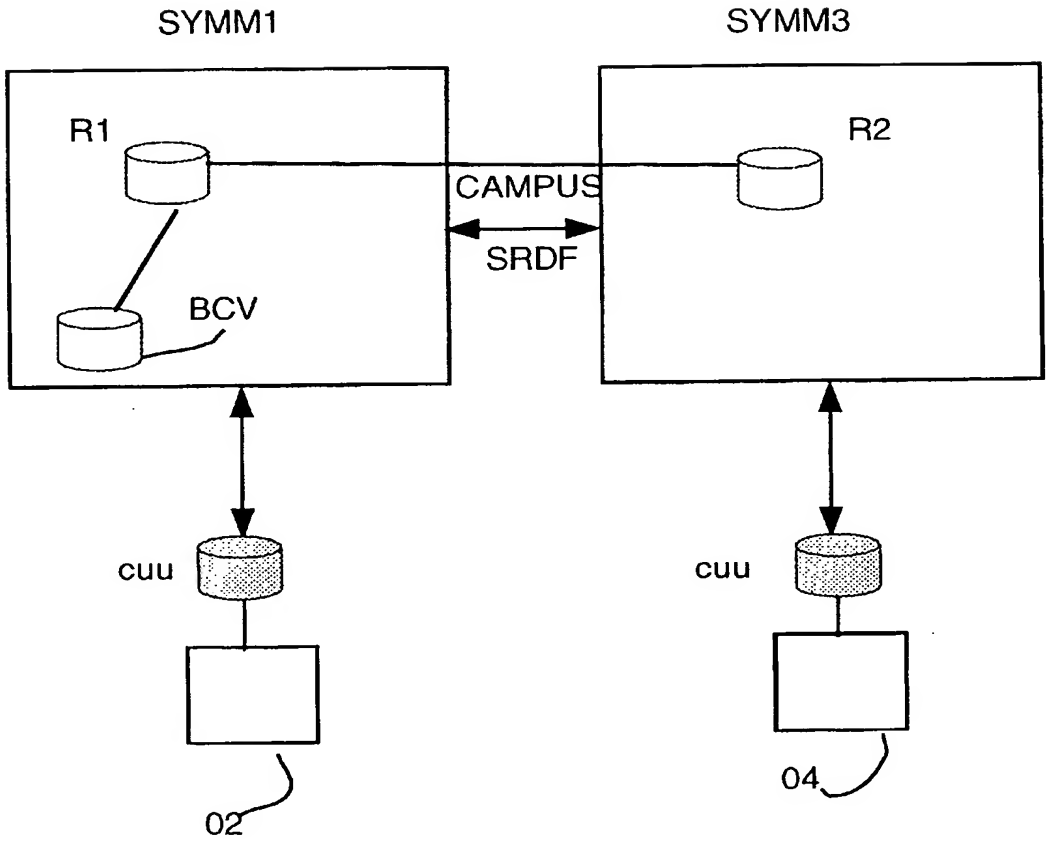


*Fig. 3a*

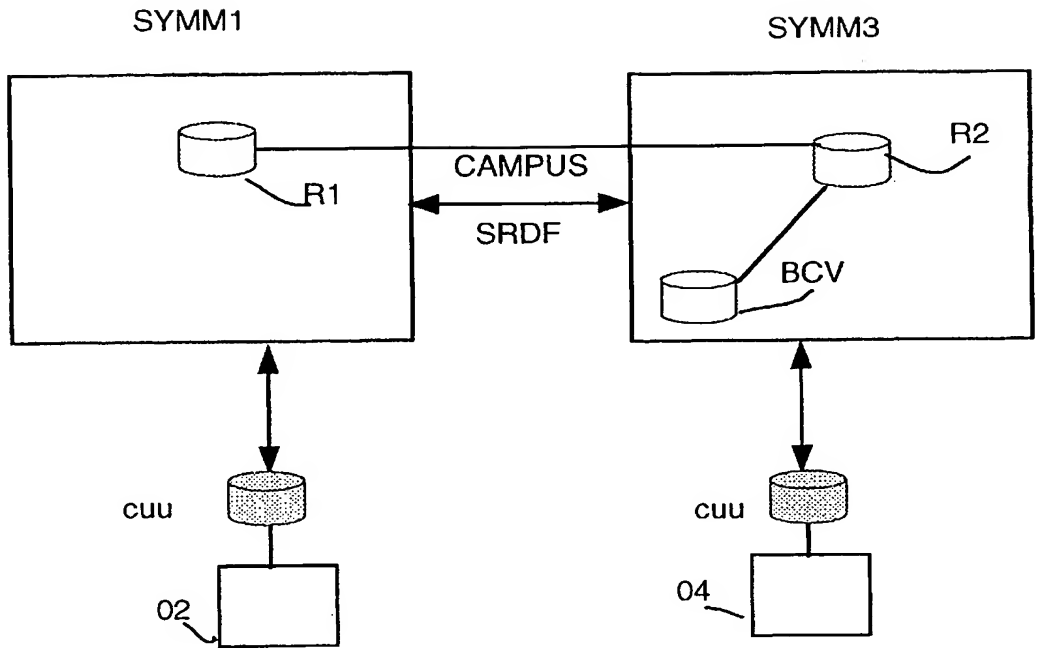

SYMM1
SYMM3
R1
CAMPUS
SRDF
R2
BCV
cuu
cuu
02
04

*Fig. 3b*

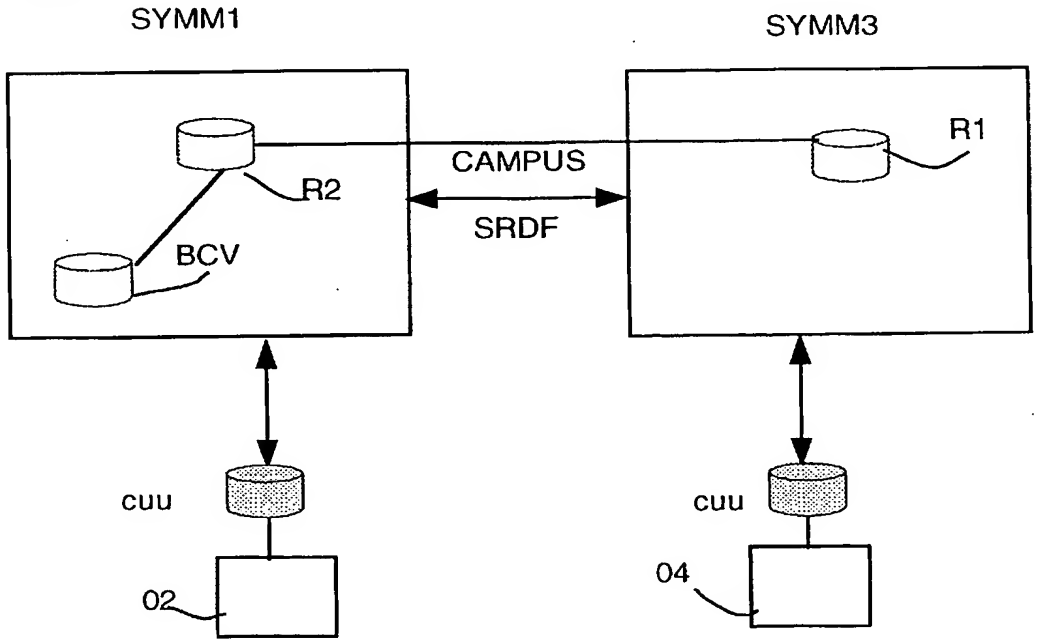

SYMM1
SYMM3
R2
BCV
CAMPUS
SRDF
R1
cuu
cuu
02
04

*Fig. 4a*


*Fig. 4b*

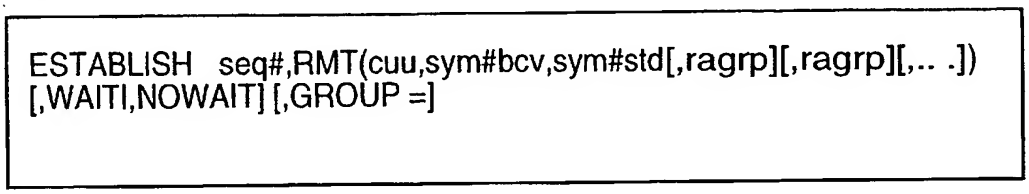



*Fig 4c*

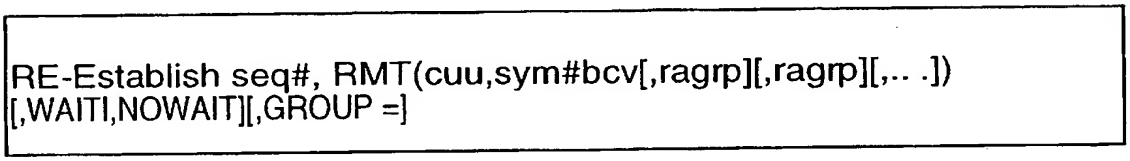



*Fig. 4d*

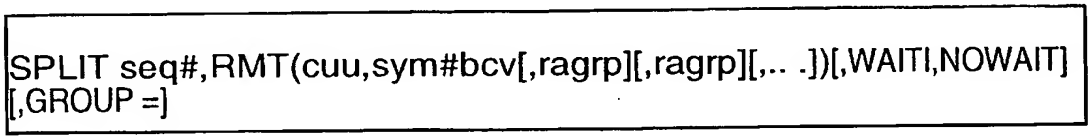



*Fig. 4e*

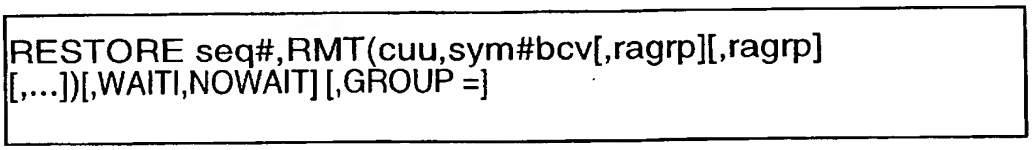


*Fig. 5*

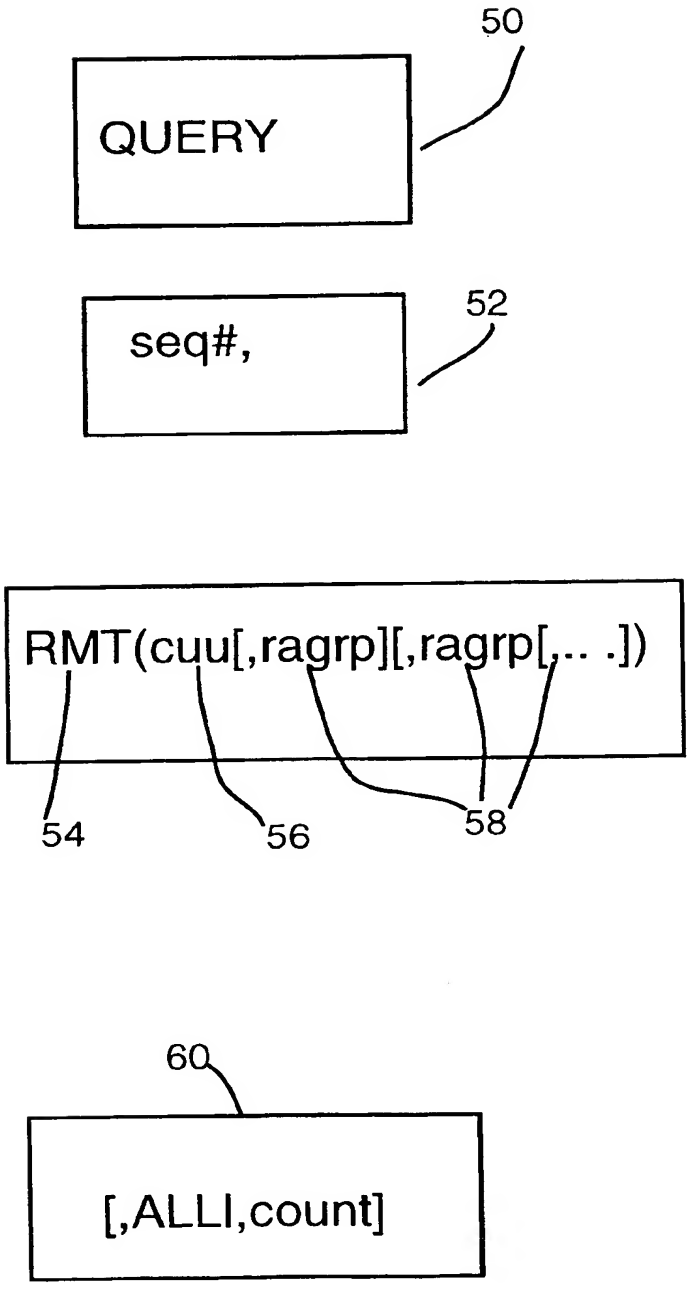


*Fig. 6a*

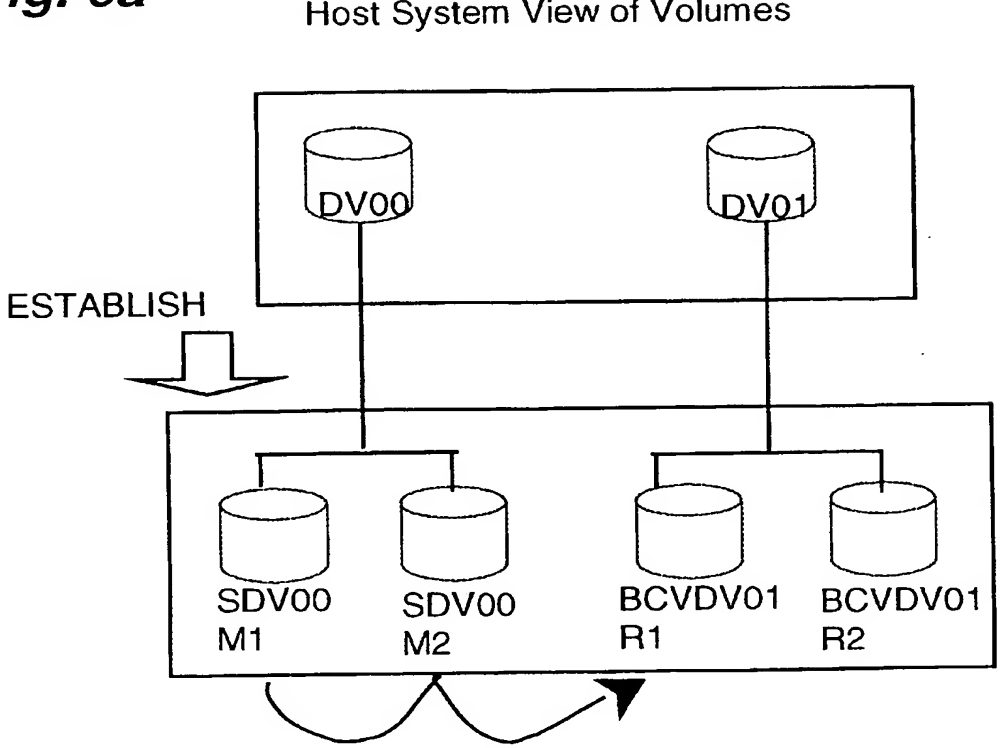

Host System View of Volumes
DV00
DV01
ESTABLISH
SDV00
M1
SDV00
M2
BCVDV01
R1
BCVDV01
R2

*Fig. 6b*

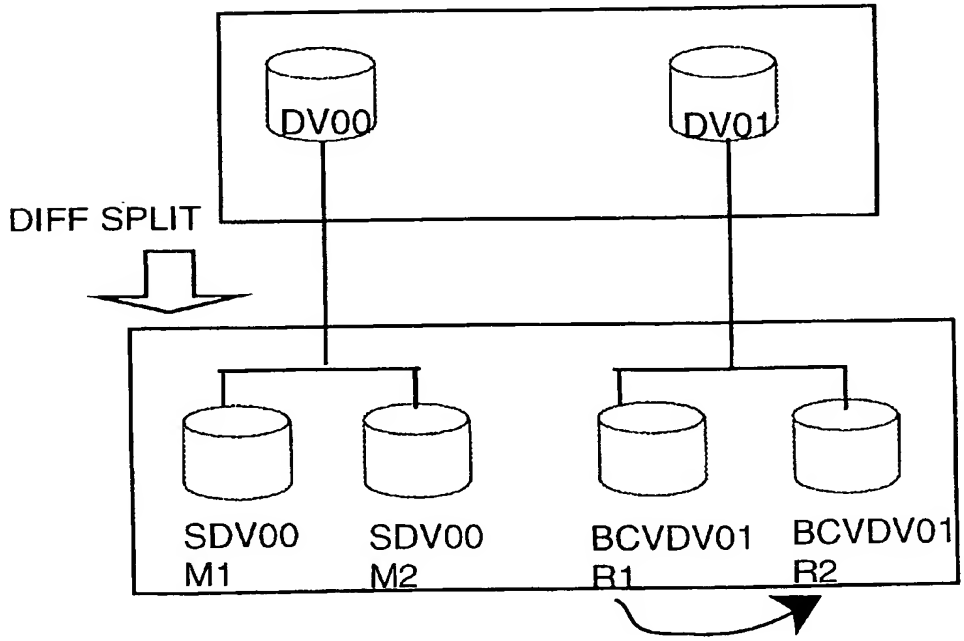

Host System View of Volumes
DV00
DV01
DIFF SPLIT
SDV00
M1
SDV00
M2
BCVDV01
R1
BCVDV01
R2

*Fig. 6c*

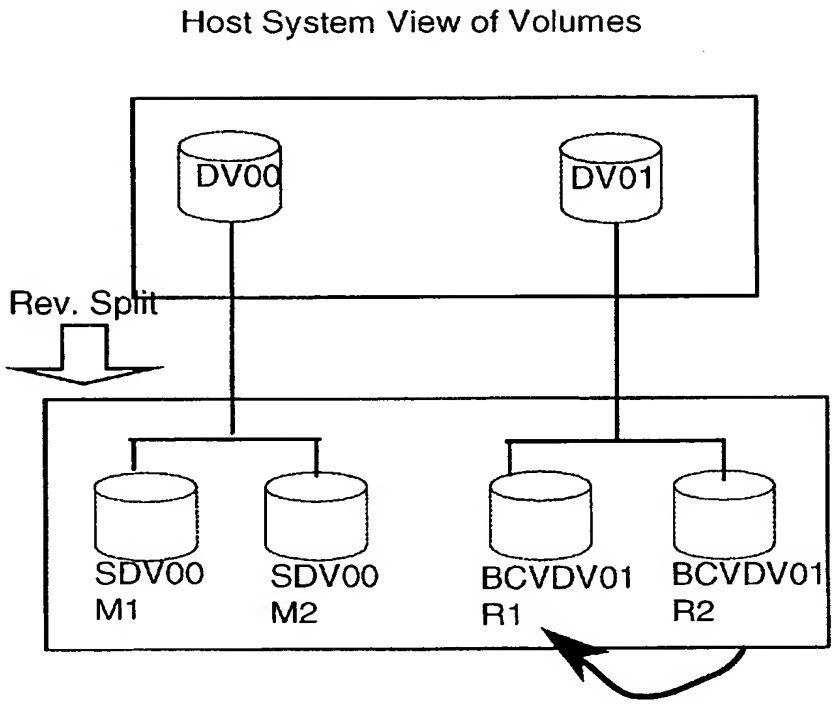

Host System View of Volumes
DV00
DV01
Rev. Split
SDV00
M1
SDV00
M2
BCVDV01
R1
BCVDV01
R2

*Fig. 6d*

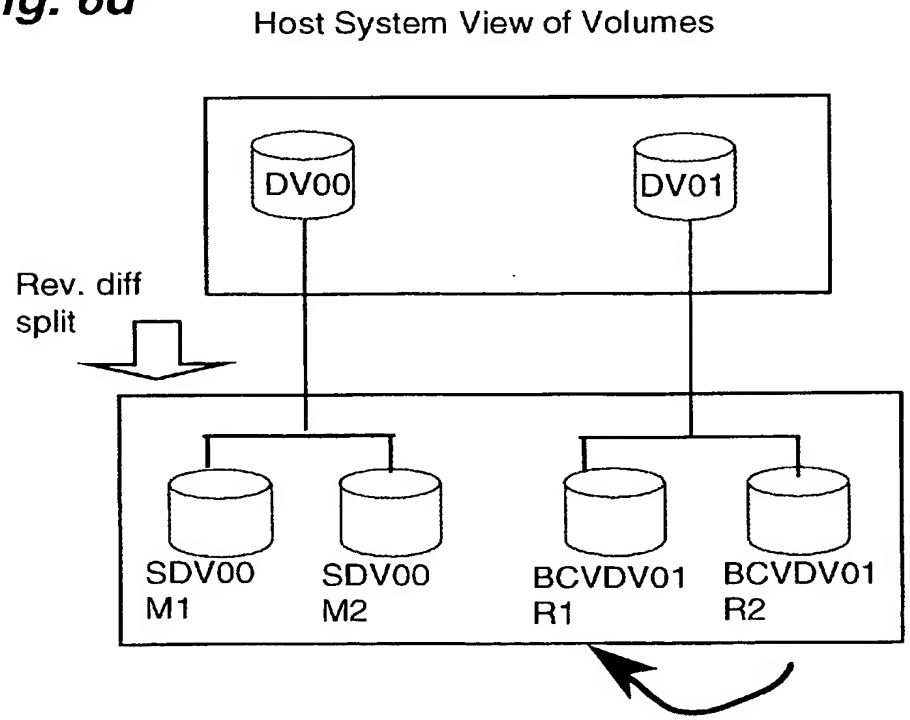

Host System View of Volumes
DV00
DV01
Rev. diff
split
SDV00
M1
SDV00
M2
BCVDV01
R1
BCVDV01
R2

*Fig. 7a (Prior Art)*

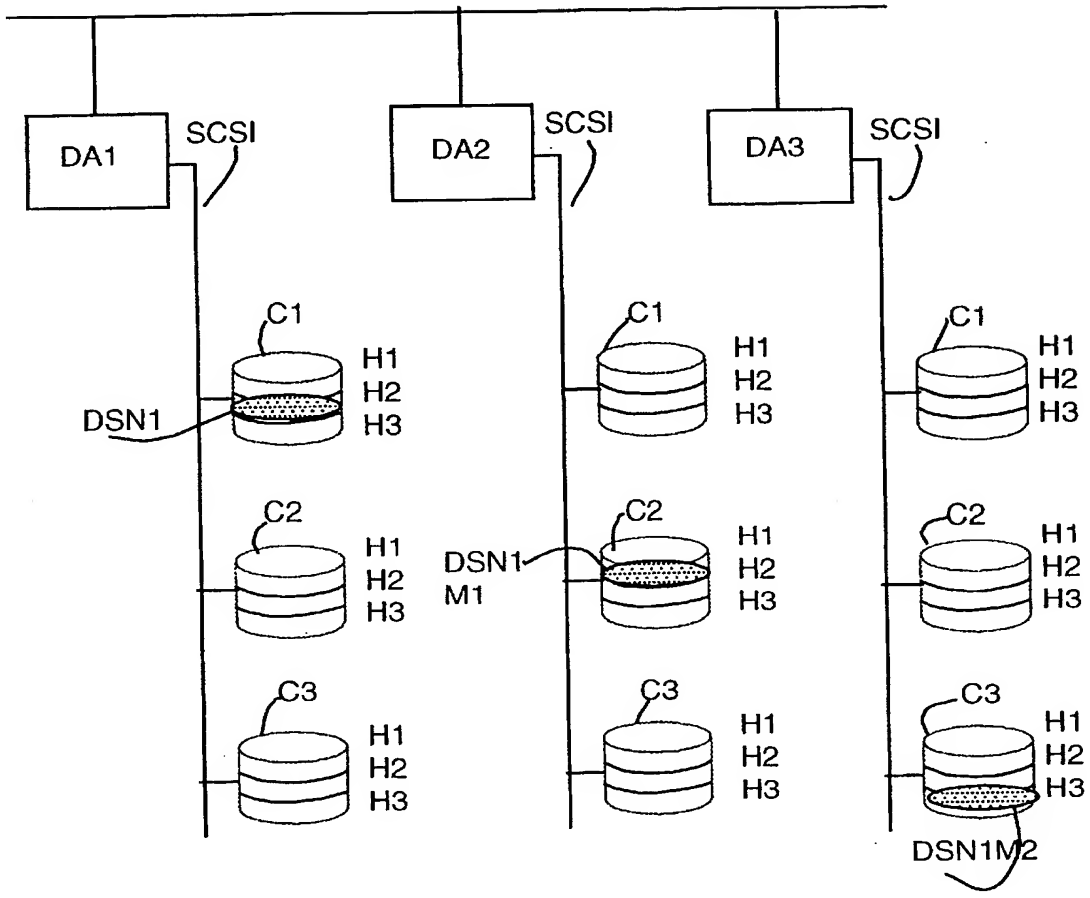



*Fig. 7b (Prior Art)*

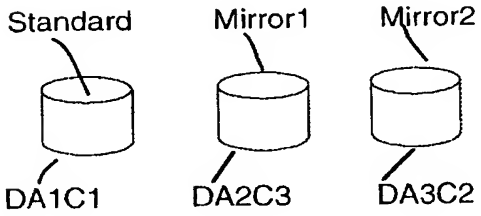


*Fig. 7 c( Prior Art)*

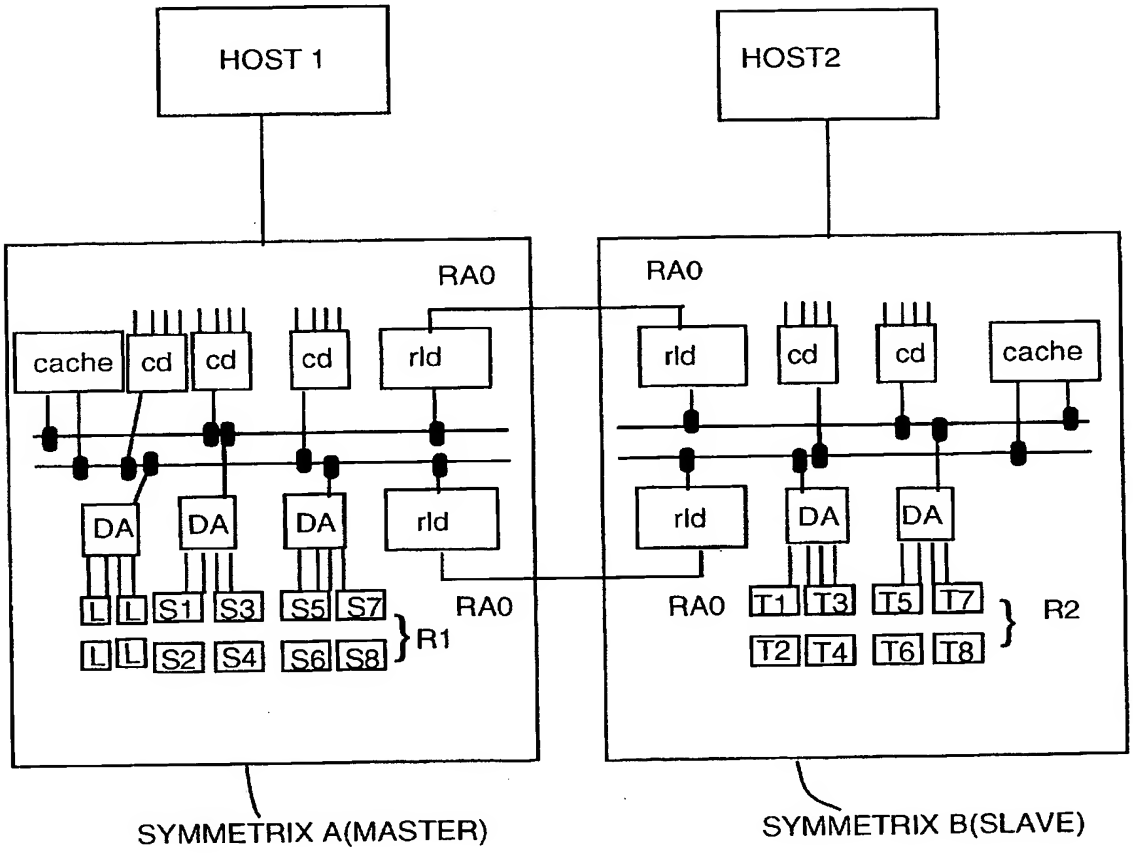


*Fig. 7d (Prior Art))*

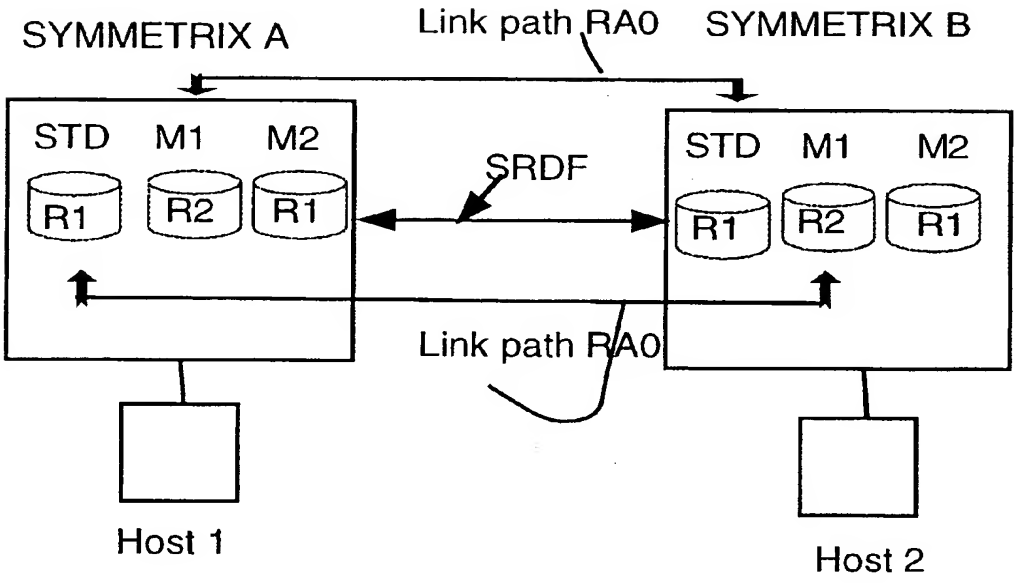


*FIG. 8 (Prior Art)*

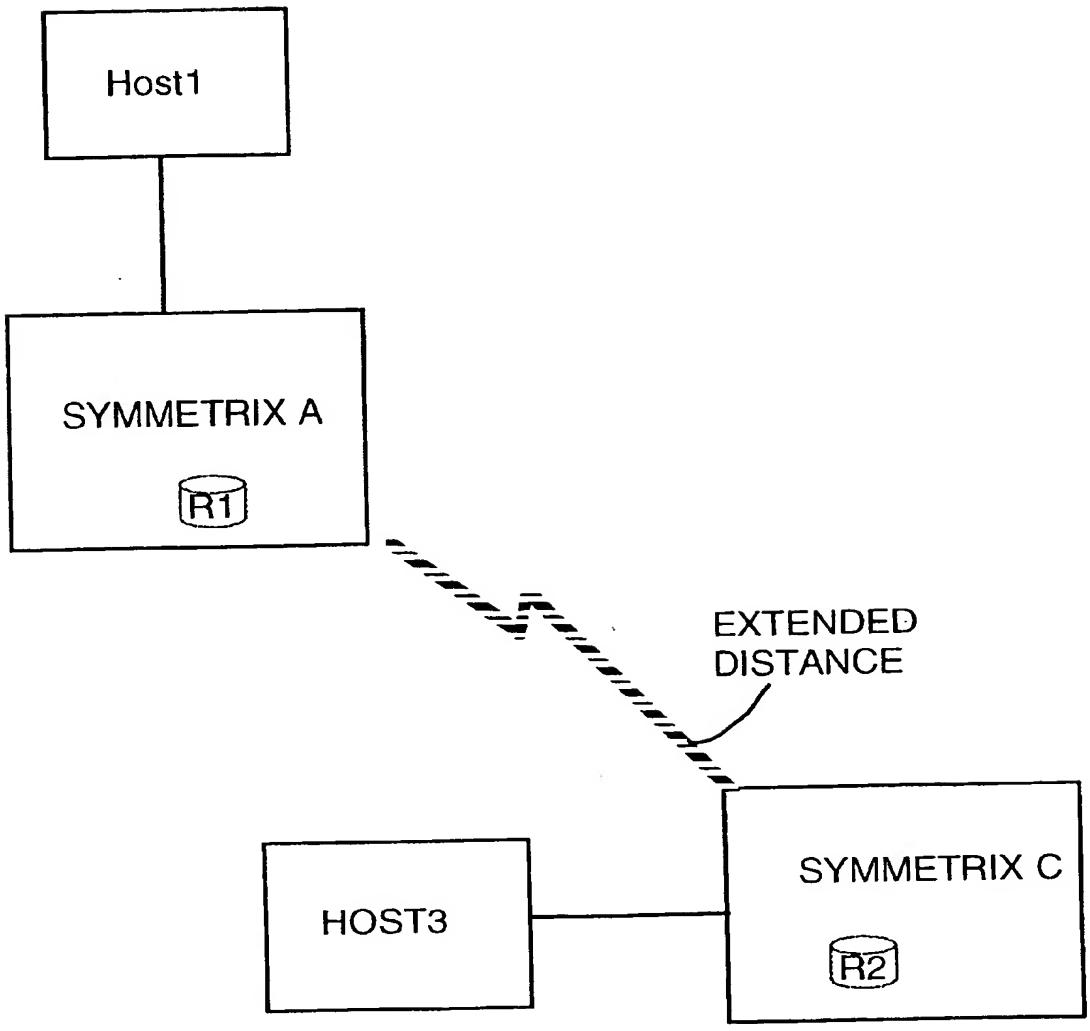


*Fig. 9*

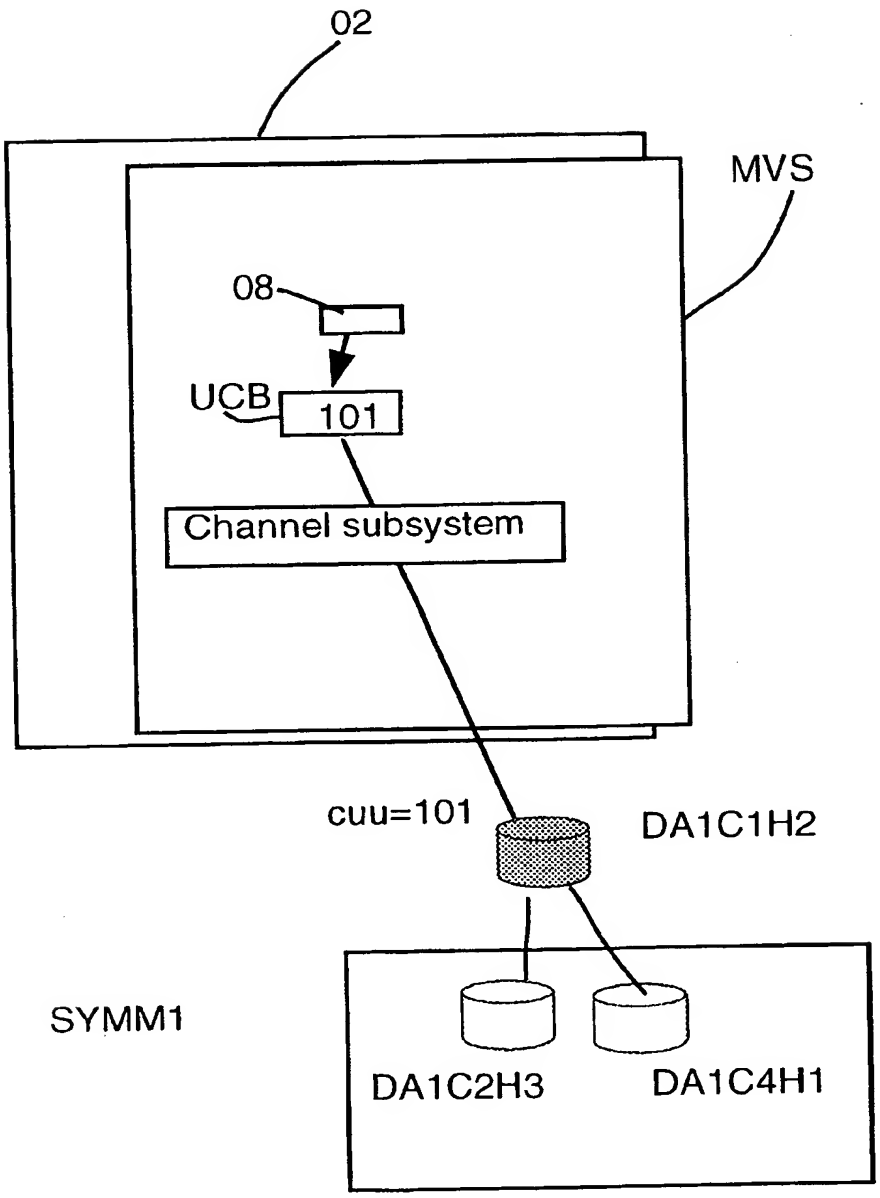

02
MVS
08
UCB
101
Channel subsystem
cuu=101
DA1C1H2
SYMM1
DA1C2H3
DA1C4H1

*Fig. 10*

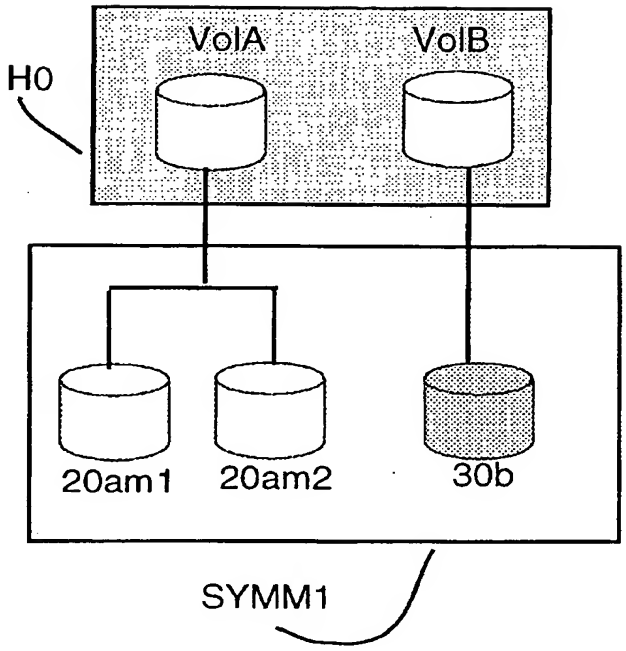

VolA
VolB
H0
20am1
20am2
30b
SYMM1

*Fig. 11*

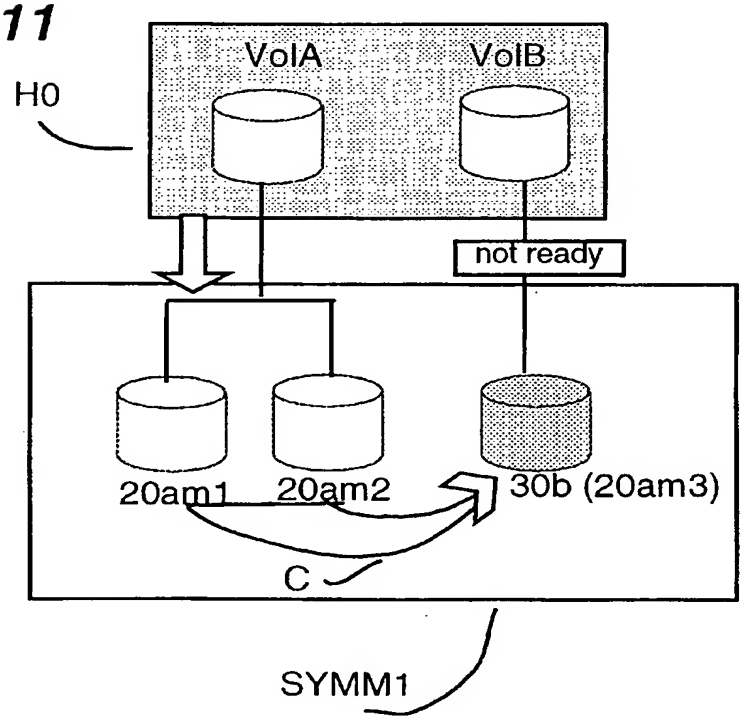

VolA
VolB
H0
not ready
20am1
20am2
30b (20am3)
C
SYMM1

*Fig. 12a*

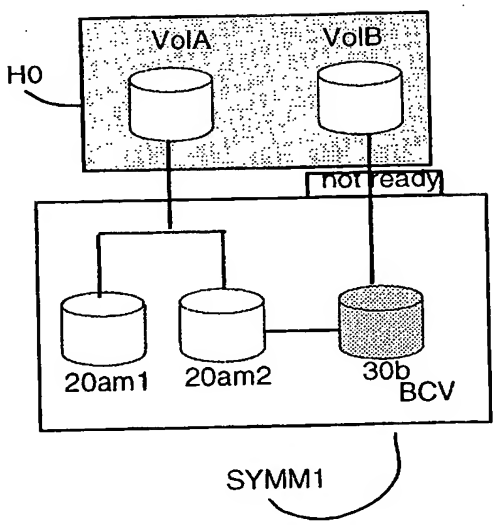



*Fig. 12b*

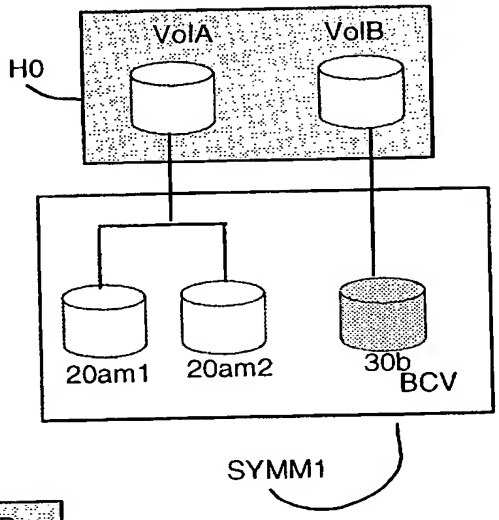



*Fig. 13*

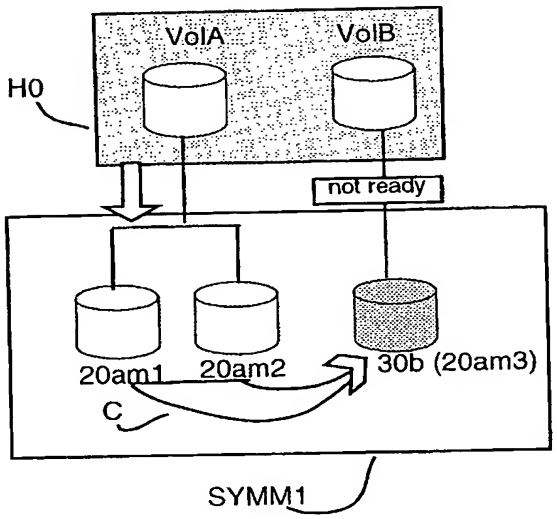


*Fig. 14*

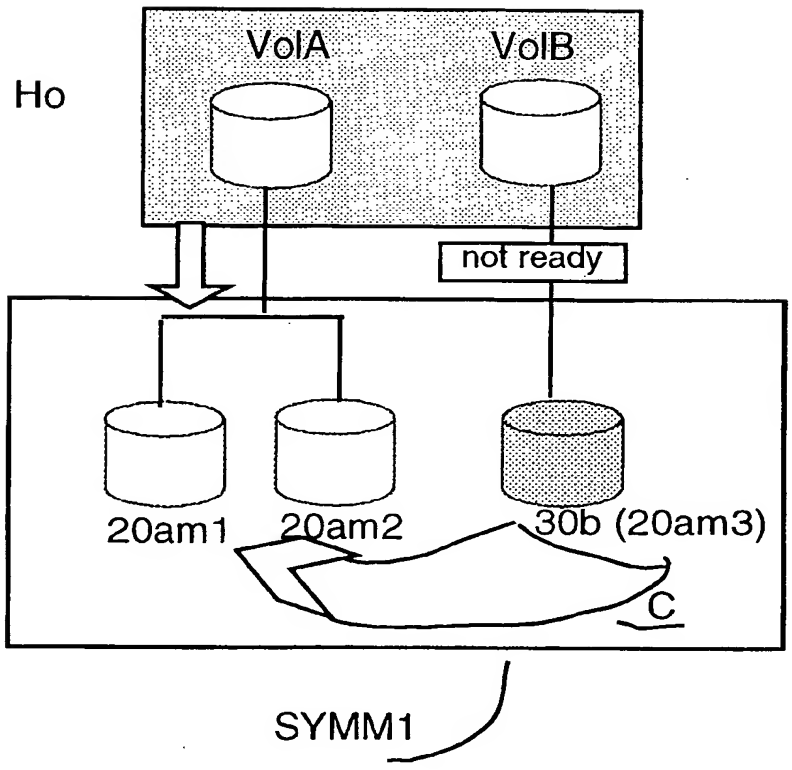


*Fig. 15*

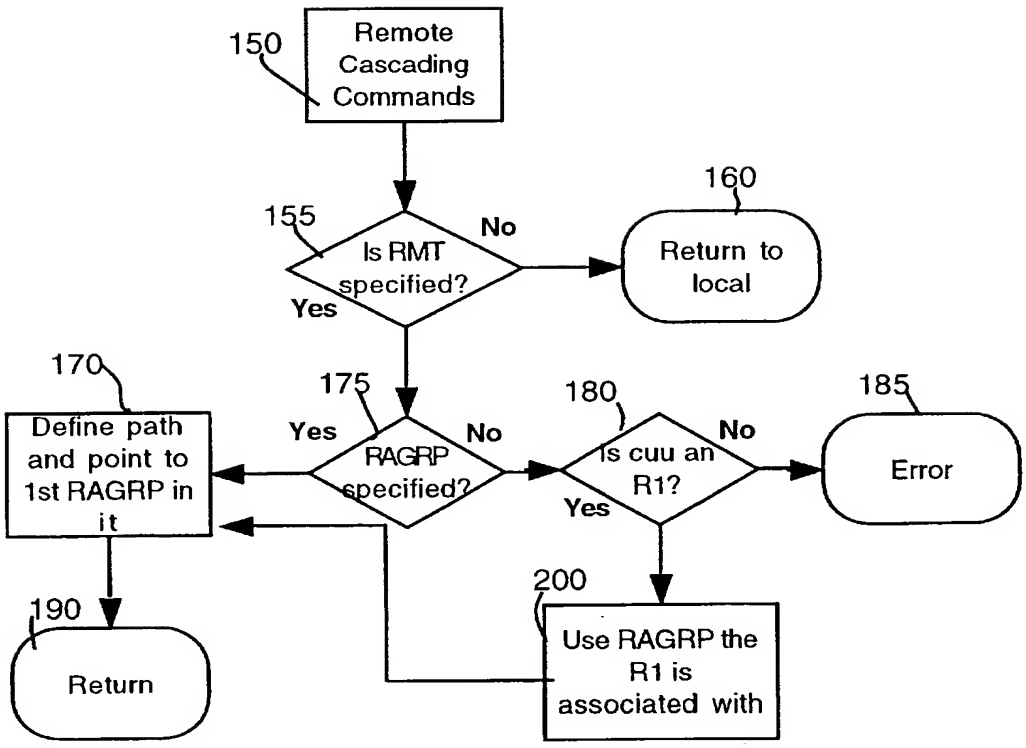

150
Remote Cascading Commands
155
Is RMT specified?
No
Yes
160
Return to local
170
Define path and point to 1st RAGRP in it
175
RAGRP specified?
Yes
No
180
Is cuu an R1?
No
Yes
185
Error
190
Return
200
Use RAGRP the R1 is associated with

*Fig. 16*

Table

| | Addr | Source | Target | RAGRP |
|---|---|---|---|---|
| Line1 | 1 | R1-26 | R2-19 | 0 |
| Line2 | 1 | R1-27 | R2-18 | 1 |

# HOST SYSTEM FOR REMOTE CONTROL OF MASS STORAGE VOLUMES USING CASCADING COMMANDS

## BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to the field of mechanisms for allowing remote control to occur between host computers and a plurality of mass storage business continuance volumes and more particularly to cascading commands for issuance by a host system to collect information about and transmit control commands to volumes at one or more levels away from the host in the system.

2. Background

Mass storage systems have become increasingly cost effective for critical business systems. Advances such as Redundant Arrays of Independent Disks (RAID) technologies and Hierarchical Storage Management (HSM) systems have greatly improved the reliability of mass storage by providing a number of different redundancy features. Additionally, HSM systems such as the SYMMETRIX™ systems that are commercially available from the Assignee of the subject application provide disaster recovery facilities such as Symmetrix Remote Data Facilities (SFDF). These allow a SYMMETRIX™ 5xxx system located at one site to maintain a continuous copy or mirror of the data at the logical volume level in other SYMMETRIX™ systems located in physically separate sites. FIG. 7*a* (Prior Art) illustrates a redundancy technique used in SYMMETRIX™ systems to provide mirroring, RAID configurations, and other forms of redundant disk storage. As seen in FIG. 7*a* (Prior Art) disk adapters DA1, DA2 and DA3 are connected over small computer storage interface (SCSI) buses to physical disk drives C, such as C1, C2 and C3 on disk adapter DA1. In SYMMETRIX™ systems, in some implementations, a physical disk C1 is divided into three logical disks, called H1, H2, and H3.

To illustrate this, assume a typical physical disk connected to a mainframe computer contains 2000 cylinders. In an HSM system such as SYMMETRIX™ systems, shown in FIG. 7*a* (Prior Art) using disks C which are larger in capacity than the typical disks, the larger disks C can be logically divided into smaller logical units. If the disks C in this example hold 6000 cylinders, this physical disk C has the capacity of three typical disks. Each logical disk H, in this example, would be the equivalent of one typical disk.

Still in FIG. 7*a* (Prior Art), if a typical disk contains a single large file or dataset named DSN1, mirroring redundancy techniques used in an HSM such as SYMMETRIX™ systems, can create copies of this dataset DSN1 on disk adapters DA2 and DA3. In the example here, the first standard copy of DSN1 is physically located on disk adapter DA1, disk C1, at logical disk H2. The first mirror copy, DSN1M1 is located physically on disk adapter DA2, at physical disk C2, logical disk H1. The second mirror copy DSN1M2 is located on disk adapter DA3, physical disk C3, logical disk H3.

In FIG. 7*b* (Prior Art), a more abstract way of thinking about mirroring or redundancy is shown. If the HSM system allows three mirrors for a standard disk, the HSM might have disks configured as shown here—the standard disk for data DSN1, is allocated to disk adapter DA1, physical disk C1. The first mirror, Mirror1, is assigned to disk adapter DA2, physical disk C3, and so on. In SYMMETRIX™ systems, the combination of disk adapter DA, physical disk C and logical disk H, is resolved into a SYMMETRIX™ device number. In this example, the SYMMETRIX™ system synchronizes the mirrors in a transparent manner. When the data from disk adapter DA1, physical disk C1 has been copied to mirror1 and mirror 2, the devices are considered synchronized.



Now turning to FIG. 7*c* (Prior Art), disk adapters DA are shown as they might be configured in Symmetrix Remote Data Facilities (SRDF) systems for disaster recovery. As seen in FIG. 7C (Prior Art), a SYMMETRIX A(MASTER) system has been configured in a unidirectional SRDF campus solution with SYMMETRIX B(SLAVE) system.

When the SRDF features are used, a SYMMETRIX™ system includes not only cache memory, channel directors CD and disk adapters DA, but remote link directors RLD. Within each SYMMETRIX™ unit, three volume types may be defined: local (L), source (R1) and target (R2). Local volumes L are volumes unique to that SYMMETRIX™ unit. They are only accessible to hosts attached to that SYMMETRIX™ unit (in this example, HOST 1.)

Still in FIG. 7*c* (Prior Art), source volumes R1 are logical volumes that reside on a SYMMETRIX™ unit with the SRDF features activated, so that the data on source volumes R1 is mirrored or copied to respective target volumes R2 on another SYMMETRIX™ unit (in this example, HOST 2). The target volumes R2 are located on one or more separate SYMMETRIX™ units in an SRDF configuration.

As seen in FIG. 7*c* (Prior Art), a path is established by the remote link directors RLD to allow data to be mirrored. The paths shown here are labeled remote access group 0, or RA0.

Turning now to FIG. 7*d* (Prior Art), a bidirectional SRDF configuration is shown. Host 1 is logically in communication with standard volume std, in this example. As mentioned above, SYMMETRIX™ systems would normally (in the absence of SRDF features) establish some mirroring for an ordinary volume. In this case, mirrors M1 and M2 in SYMMETRIX™ A might be established for standard volume std. The SRDF feature takes this mirroring one step further. Instead of creating a mirror of SRDF standard source volume R1 on SYMMETRIX™ A, the SRDF features, using the remote link directors RLD, assign a remote mirror M1 in SYMMETRIX™ B. In other words, source volumes R1 are standard volumes that reside on a SYMMETRIX™ unit, with the data on those volumes mirrored to respective target R2 volumes on another SYMMETRIX™ unit, here SYMMETRIX™ B. If the source volume R1 fails, the SYMMETRIX™ A will transparently access the data on the corresponding target volume R2 in SYMMETRIX™ B. When the failing volume is replaced, the remotely mirrored pair is re-synchronized automatically as a background operation, using the data in the target volume R2.

Still in FIG. 7*d* (Prior Art), target R2 volumes are a copy of the source R1 volumes in another SYMMETRIX™ unit. A target volume R2, typically has a default configuration mode of "read-only" to any host with access to the SYMMETRIX™ unit in which it resides. In this example, mirror M1, in SYMMETRIX™ B, which is the remote mirror to the source volume R1 in SYMMETRIX™ A, would be made "not-ready'" to Host 2. Normally, writes to the target volume R2 on Host 2, occur via the link paths created by the remote link director. However, if the source volume R1 on SYMMETRIX™ A fails, SYMMETRIX™ A will transparently access the data on its corresponding target R2 volume (here, M1 in SYMMETRIX™ B)

A target volume R2 typically has a default configuration mode of "read-only" to any host with access to its SYM-

METRIX™ unit, in this example, HOST 2. To enable disaster recovery, the remote link directors RLD in a SYMMETRIX™ unit, create link paths RA0 with the SYMMETRIX™ unit containing the target volumes R2. As data is written to a source volume R1 by HOST 1, the remote link director RLD in SYMMETRIX™ A automatically writes a copy of that data over link path RA0 of the SRDF connection to the corresponding target volume R2 on the designated SYMMETRIX B system. Thus, if a source volume R1 fails for any reason, the remote link director RLD will transparently access the data on the corresponding target volume R2 on the designated target SYMMETRIX B system over the link paths RA0 and transmit that to HOST 1.

New writes to the failed source volume R1 in SYMMETRIX™ A accumulate as invalid tracks in the cache of SYMMETRIX™ A—the unit containing the source volume R1. When the failing source volume R1 is replaced, the remotely mirrored pair is re-synchronized automatically as a background operation within the SYMMETRIX™ unit, using the data in the appropriate target volume R2.

FIG. 8 (Prior Art) shows an extended distance implementation of the SRDF features. So far, the above discussion dealt primarily with an SRDF CAMPUS connection, which usually involves physical proximity, such as units located in two buildings on a college campus. In extended distance configuration shown in FIG. 8 (Prior Art), the source volume R1 on SYMMETRIX™ unit SYMMETRIX A is being automatically and transparently copied to target volume R2 on SYMMETRIX C over the SRDF EXTENDED DISTANCE connection. The SRDF EXTENDED DISTANCE solution allows source volume R1 to be automatically copied to target volume R2 on SYMMETRIX C, which may be located thousands of miles away. Thus, if a disaster such as a flood destroys all the hosts and SYMMETRIX™ units located at a campus site, the application can still be run from the remote, extended distance site represented here by SYMMETRIX C. Sites can be as much as 37.5 miles (60 km) apart from each other in a "campus" solution, or in an extended distance solution over 37.5 miles (60 km) apart using T3 or E3 or similar high speed links.

These disaster recovery remote sites, whether campus or extended ones, are mirrors or copies of data stored on the SYMMETRIX™ systems physically located near the host computer. As described above, the copying is done by the respective SYMMETRIX™ systems automatically, once all appropriate SYMMETRIX™ systems have been configured for the SRDF feature. This means the host(s) and local SYMMETRIX™ systems at each site are able to operate as efficiently as if the remote copying were not occurring. That is, the continuous copying to the remote sites is not "logically visible" to the host computers at the various sites, nor does it interfere with processing at the host. Hence, a host computer, is not able to send commands that effect changes to the remote sites. This is true even if the remote sites are located only one step away in a campus solution. If the host wishes to effect changes, either the CAMPUS or the extended sites, there was heretofore no way of doing this directly at the local site.

It is an object of this invention to enable a host computer to cause volumes to be managed remotely.

It is another object of the present invention to enable a host computer to issue commands to effect changes in volumes at remote SRDF sites.

Still another object of the present invention is to enable a host computer to collect information about volumes at remote SRDF sites.

## SUMMARY OF THE INVENTION

These and other objects are achieved by a host system for remote control of mass storage volumes using cascading commands which collect information indirectly about a stream of linked volumes attached to mass storage volumes attached to other hosts or located at physically separate sites. A host computer program issues the cascading commands which ask the locally communicating mass storage system to return information which can be used to identify one or more levels of remote mass storage systems. Once a mass storage system at a given level has been identified, commands can be sent by the host through the locally communicating mass storage system to cause actions to occur at the identified remote level, whether or not there are multiple intervening levels of remote mass storage systems. In the embodiment shown, a host computer can query, establish, split, re-establish, copy and restore business continuance volumes at any level in a chain of local and remote mass storage system sites. Those skilled in the art will appreciate that other commands could be implemented as well to effect changes in the volumes at remote sites.

It is an aspect of the present invention that it enables host commands to cascade over one or more intervening levels of mass storage systems at remote sites to take effect at a particular designated site.

Still another aspect of the present invention is that it enables a host to monitor the operations being controlled at the designated remote site.

Still another aspect of the present invention is that it can significantly improve the remote site's ability to be effective as a business continuance site for disaster recovery.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of the present invention.

FIG. 2 is a block diagram illustrating establishment of a local business continuance volume.

FIG. 3*a* is a block diagram illustrating use of the present invention with two linked sites and the establishment of a business continuance volume at a remote site.

FIG. 3*b* is a block diagram illustrating another use of the present invention.

FIGS. 4*a*–4*e* block diagrams of the syntax of cascading remote business continuance volume commands of the present invention.

FIG. 5 is a detailed view of cascading business continuance volume command of the present invention.

FIG. 6*a* is a block diagram of an establish command using the cascading commands of the present invention.

FIG. 6*b* is a block diagram of a differential split command using the cascading commands of the present invention.

FIG. 6*c* is a block diagram of a reverse split command using the cascading commands of the present invention.

FIG. 6*d* is a block diagram of a reverse differential split command using the cascading commands of the present invention.

FIG. 7*a* (Prior Art) is a block diagram of the physical and logical disk structures of an HSM system.

FIG. 7*b* (Prior Art) is a block diagram of disk mirroring.

FIG. 7*c* (Prior Art) is a block diagram of disaster recovery features linking two physically separate sites.

FIG. 7*d* (Prior Art) is a block diagram of remote disk mirroring using SYMMETRIX™ SRDF features.

FIG. 8 (Prior Art) is a block diagram of an extended distance configuration of a SYMMETRIX™ system.

FIG. **9** is a block diagram of the logical flow of data between a host operating system and mass storage subsystems.

FIG. **10** is a block diagram of simple mirroring in a mass storage subsystem.

FIG. **11** is a block diagram of a mass storage subsystem in which a local business continuance volume is being established.

FIG. **12***a* is a block diagram of a mass storage subsystem in which a local business continuance volume has been established.

FIG. **12***b* is a block diagram of a mass storage subsystem in which a local business continuance volume has been split from its standard volume.

FIG. **13**, is a block diagram of the re-establishment of a business continuance volume in a local system.

FIG. **14** is a block diagram of a restore command taking effect in a BCV volume in a local system.

FIG. **15** is a flow diagram of the present invention.

FIG. **16** is an illustrative table used with the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

In FIG. **1**, a block diagram of the present invention is shown. In this example, a host computer system, host **02**, is executing an operating system MVS, which is IBM Corporation's MVS operating system. Those skilled in the art appreciate other operating systems for a host, such as UNIX or MICROSOFT's NT™ operating systems could be used, among others. Host component software **08** allows a user to cascade commands from host **02** to a SYMMETRIX™ system such as SYMM2, at the LOCAL site, to SYMMETRIX™ SYMM5 at the CAMPUS site to SYMMETRIX™ system SYMM6 at an EXTENDED site.

Still in FIG. **1**, commands can be issued from host **02**, SYMM2 to take effect at SYMM5 and SYMM6, for example, by establishing a BCV volume (R1BCV) in SYMM5, which causes the contents of standard volume R2 in SYMM5 (which are actually the contents of R1B in SYMM2) to be copied to R1BCV. When the R1BCV volume is subsequently split off, it becomes a source volume R1BCV in SYMM5 to target volume R2 in SYMM6. Thus, the contents of disk R1B, in SYMM2 at the LOCAL site, have been propagated to disk R2 in SYMM6 at an EXTENDED DISTANCE location possibly thousands of miles away. All of the copying needed to effect this is transparent to each of the hosts at each site. In this example, the contents of disk R2 on SYMM6 at the EXTENDED location can serve as a disaster recovery backup, as of a certain time and date, for the business applications running on host **02**.

Turning briefly to FIG. **9**, an overview of the interface between host software **08** and a SYMMETRIX™ system SYMM1 is shown. Host **02** is connected to mass storage subsystem SYMM1. In the MVS environment, the operating system software MVS links computer programs to physical devices by means of a unit control block, UCB, which, in turn, maps into the MVS operating system's channel subsystem. The channel subsystem takes logical input/output requests and maps them into channel path selections to a particular device and ultimately, physical input/output signals for that device. The logic inside a SYMMETRIX™ system does the final mapping which assigns the logical device to one or more mirrors and identifies the SYMMETRIX™ device numbers for each device and mirror involved, in this case, DA1C1H2, and DA1C2H3 and DA1C4 H1. Those skilled in the art will appreciate that other operating systems have similar mechanisms for separating the logical input/output requirements from the physical as this enables much more flexibility for configuring the hardware.

U.S. patent application Ser. No. 08/842,953 filed Apr. 25th, 1997, issued as U.S. Pat. No. 6,101,497 on Aug. 8th, 2000, entitled "Method and Apparatus for Independent and Simultaneous Access to a Common Data Set", ("953") which assigned to the Assignee of the subject application, and is also hereby incorporated by reference, generally describes a way to further enhance SYMMETRIX™ systems by allowing the establishment of what are known as Business Continuance Volumes (BCV) in a SYMMETRIX™ system. In addition, co-pending U.S. patent application Ser. No. 09/039/020, filed Mar. 13, 1998, entitled "Host System for Mass Storage Business Continuance Volumes" ("020") which is assigned to the Assignee of the subject application and is also hereby incorporated by reference, generally describes a system for enabling a host computer, using an operating system such as MVS, to find the time to establish and manage such business continuance volumes in a local SYMMETRIX™ system without disrupting critical business programs.

Since the establishment of Business Continuance Volumes (BCV), as generally described in application 020, makes use of ordinary mirroring capabilities found in SYMMETRIX™ units and similar HSM systems, simple mirroring is illustrated at FIG. **10**.

In FIG. **10**, the logical volumes as the host sees them are shown in the box labeled **H0**. In this and the next few Figures, the host recognizes two logical volumes by the volume serial numbers VolA and VolB. Mass storage subsystem SYMM1 actually stores the data for these volumes on its own logical volumes. Here, mirroing has been configured for the data on a logical SYMMETRIX™ volume with a volume serial number of VolA. Mass storage subsystem SYMM1 assigns this logical volume to logical volume **20***am***1**. To create a mirror disk for it, it assigns another logical volume **20***am***2**. Mass storage subsystem SYMM1 maintains logical volumes **20***am***1** and **20***am***2** as mirror copies of each other, but host **H0** and the MVS operating system work with logical volume serial VolA as though it were the only logical volume in use.

Next, as generally described in co-pending application 020, SYMM1 uses the same mirroring techniques to establish a business continuance volume BCV **30***b*. Any mirror structure can be used, such as normal mirroring, RAID, mirroring with the Assignee's Symmetrix Remote Data Facilities (SRDF), and so on.

The establishment of a BCV volume, using the ESTABLISH BCV command is shown in FIG. **11**. Host component software causes mass storage subsystem SYMM1 to set logical volume **30***b* as "not ready" to host **H0** (and any other hosts H connected to mass storage subsystem SYMM1.) Mass storage subsystem SYMM1 then assigns BCV volume **30***b* as the next available mirror for standard volume mirrors **20***am***1** and **20***am***2**. This assignment of the BCV volume to mirror status enables mass storage subsystem SYMM1 to copy the contents of standard mirror volumes **20***am***1** and **20***am***2** to BCV volume **30***b* (**20***am***3**) as shown by the arrow C in FIG. **11**. While the copying is taking place, it is transparent to the host, which may continue to make I/O requests to logical volume VolA. For better performance,

half the data is copied to BCV volume **30***b* from disk **20***am***1** and the other half from disk **20***am***2**.

Now as is shown in FIG. **12***a*, when the copying is complete and BCV volume **30***b*(**20***am***3**)'s contents are identical to those of standard volume **20***am***1** and **20***am***2**, BCV volume **30***b* (**20***am***3**) is now synchronized with standard volumes **20***am***1** and **20***am***2** and capable of working as a business continuance device. This copying is transparent to, and not disruptive of the business application running on the host. Once a BCV volume has been synchronized by mass storage subsystem SYMM**1**, it is considered part of a BCV pair, and any new data written to the standard volume of such a pair is also written to the BCV volume of the BCV pair by mass storage subsystem SYMM**1**. The BCV volume will continue to be marked as "not ready" for any hosts H which use mass storage subsystem SYMM**1** as long as it is a part of a BCV pair.

FIG. **12***b* shows the result of a split command described in application 020. The split command allows the user to stop the mirroring operation and frees up the new copy of the data for other purposes, such as testing or restore operations. The response to a split command suspends any I/O to the BCV volume **30***b* of the BCV pair, changes its device state to ready, enables access to it from any host H, while still containing the volume serial number which is the duplicate of the standard volume.The device is still offline, after the split since it was marked offline in order to originally establish or restore the volume. Mass subsystem SYMM**1** flags any new writes to the standard volume that occur after the split (in case the BCV is re-established later) and resumes normal operations. New writes to the BCV volume are also flagged, after the split so that when a re-establish or incremental restore is done, as described in more detail in application 020, mass storage subsystem SYMM**1** can compare the changes to both volumes and refresh the tracks that have changed.

FIG. **13** shows the process of re-establishing a BCV pair, in which only the changes that have been made to standard volumes **20***am***1** and **20***am***2** are copied to volume **30***b*. Any tracks that had been changed on volume **30***b* as a result of its use after a split, are overwritten by the data on the corresponding track on the standard volumes.

FIG. **14** illustrates the restore command which restores standard volumes **20***am***1** and **20***am***2** with all the data from the BCV volume, now seen as volume **30***b*(**20***am***3**). A restore is done by setting the BCV volume as the next available mirror of the standard device, and copying its data to both volumes **20***am***1** and **20***am***2**, overwriting the data present on those devices.

Returning to FIG. **1**, when EMC Corporation's SYMMETRIX™ systems, such as SYMM**1** and SYMM**2**, are installed locally and connected to a host, they, in effect, replace the simple physical devices known to the operating system MVS as MVS devices identified by an MVS device number cuu, which identifies an MVS device on a particular controller, with sophisticated HSM mirroring and redundancy systems. Thus, when the user is sending commands to MVS device number cuu, SYMMETRIX™ host component software **08** actually sends physical input and output signals through the operating system MVS's channel subsystem to the SYMMETRIX™ systems. In this example, MVS device number **101** might be mapped to a SYMMETRIX™ device on SYMM**1**, which identifies a disk adapter DA and a physical disk C.

As seen in FIG. **1**, host **02** and SYMMETRIX™ systems SYMM**1** and SYMM**2** are physically together in a LOCAL configuration. Typically, LOCAL connections are limited to a specified area defined by cable capacities which are usually equivalent to the confines of one building or floor of a building.

Using the Assignee's SRDF remote disaster recovery features described above, a local SYMMETRIX™ system SYMM**1** can be linked by SRDF, as shown in FIG. **1**, to a physically separate SYMMETRIX™ system located elsewhere inside a CAMPUS. As mentioned above, CAMPUS sites are those which are within 37.5 miles of the first SYMMETRIX™ system. If the company using the mass storage systems and disaster recovery features of the SYMMETRIX™ system is a nationwide or international company, then a SYMMETRIX™ system SYMM**5** at CAMPUS can also be linked to a SYMMETRIX™ system SYMM**6** at an EXTENDED location, which may be thousands of miles away.

SYMMETRIX™ systems in CAMPUS and EXTENDED locations are able to communicate with each other when the SRDF link paths such as remote access groups RA**0**, RA**1** and RA**0** have been established by EMC Corporation's SRDF features. However, heretofore, a host **02**, was only able to communicate with SYMMETRIX™ systems near it in location LOCAL. Commands that would have previously only been possible to send to the LOCAL SYMM**1** and SYMM**2** shown in FIG. **1**, can now be sent to remote devices at SYMMETRIX™ systems SYMM**5** and SYMM**6** at CAMPUS and EXTENDED locations, using the present invention.

In the past, EMC Corporation's SRDF facility has linked such local systems SYMM**1** and SYMM**2** to remote sites at campuses or at extended locations, but again, with the goal of keeping all the activity transparent to hosts such as host **02**. Thus, operating system MVS in host **02** of FIG. **1**, including the SYMMETRIX™ systems host component software **08**, would only have been aware of immediate SRDF status in its vicinity, but was unable to effect changes in the SRDF volumes and was unaware of the status of any SRDF devices downstream in the stream of linked SRDF devices and unable to effect any changes in them. In the present invention, this is changed, so that the status of devices at remote CAMPUS or EXTENDED sites, or anywhere in the stream can be determined by host **02**'s operating system using the SYMMETRIX™ systems host component software **08** of the present invention, and commands can be sent downstream to take effect one or two or more SRDF locations away. In effect, as seen in FIG. **1**, with the present invention, commands can "skip over" or "pass through" a locally attached SYMMETRIX™ system, and a CAMPUS SYMMETRIX™ system, to take effect at an EXTENDED SYMMETRIX™ system.

Turning briefly now to FIG. **5**, an illustrative command for collecting and managing information remotely is shown. In this example, a Query command **50** is issued by host component software **08** to MVS device cuu. The sequence number parameter functions as described in the above-referenced co-pending 020 patent application (a sequence number can be a decimal number from 1 to 128 which indicates in what order the specified command is to be executed. All actions on the same sequence level are executed in parallel.) The remote features of this command are shown at RMT **54**, cuu **56** and ragrp **58**. RMT **54** parameter specifies that the query is to gather information on the remote devices in an SRDF disaster recovery configuration. Ragrp **58** specifies the RA group through which the user wishes to perform a remote operation. For a remote command, where a remote access group (ragrp), has not

been specified, the cuu **56** parameter is interpreted by SYMMETRIX™ systems to refer to source devices (known as R1 devices) for SRDF functions. Where a remote access group ragrp **58** is known, this access group can be specified. As seen here in FIG. **5**, by the syntax of the command, the ragrp **58** operand is optional and can be positional. Thus, if the user wished to query the status of R2BCV volume in SYMM6, of FIG. **1**, such a query might appear as:

Query **1**,RMT (**202**,**1**, **0**,ALL), where cuu is MVS device number **202** in SYMM2, 1 refers to ragrp RA**1** between SYMM2 and SYMM5, and 0 refers to ragrp RA**0** between SYMM5 and SYMM6.

Turning briefly to FIG. **7***c* (Prior Art), source volumes R1 are the set of volumes labeled here as S1–S8. When the SRDF feature is activated, a remote link director RLD treats these volumes as a remote access group, here RA**0**. The SRDF feature links the source volumes R1 to target volumes R2 in SYMMETRIX B(SLAVE) shown here as volumes T1–T8), under the heading of remote access group RA**0**.

Turing back to FIG. **1**, it can be seen that SYMMETRIX™ system SYMM1 contains a source volume R1 which is linked to remote SYMMETRIX™ system SYMM3, in the CAMPUS complex by SRDF over remote access group RA**0**. SYMMETRIX™ system SYMM2, at physical location LOCAL, contains two source volumes R1A and R1B, which are linked to SYMMETRIX™ systems SYMM4 and SYMM5 by SRDF over remote access groups RA**0** and RA1. Note that in this configuration, SYMMETRIX™ system SYMM3 is referred to as remote access group **0** (RA**0**), in relation to SYMM1. SYMMETRIX™ system SYMM4 is referred to as remote access group **0** (RA**0**), in relation to SYMM2, while SYMMETRIX™ system SYMM5 is referred to as RA**1**, in relation to SYMM2. This latter SYMMETRIX™ system SYMM5, is, in turn, linked by SRDF over remote access group RA**0** to a SYMMETRIX™ system SYMM6 at an EXTENDED location. SYMMETRIX™ system SYMM6 is therefore a remote access group RA**0** to SYMMETRIX™ system SYMM5. The configuration shown in FIG. **1** is for illustrative purposes only. Other configurations are possible.

Still in FIG. **1**, by use of the present invention as well as the BCV features described in co-pending applications 953 and 020, the mass storage volume SYMMETRIX™ system SYMM6 contains remotely located copies of the data on source volume R1B in SYMMETRIX™ system SYMM2. The copies residing on SYMMETRIX™ system SYMM6 may actually have been made from the intervening level source volume R1BCV in SYMMETRIX™ system SYMM5. When SRDF is used to link SYMMETRIX™ systems, each system knows its place in the chain. Thus, the present invention enables a user to establish a remote BCV volume, R1BCV in SYMM5. Heretofore, a user could only establish a BCV volume on a locally configured SYMMETRIX system. Once the remote BCV volume has been established, it becomes a mirror of the target volume R2, on SYMM5, which, in turn, by virtue of the SRDF functions, was already a mirror of the source volume R1B in SYMM2. If the user wishes to make yet another remote copy, the user uses the remote split command to split R1BCV off from target volume R2 in SYMM5, makes R1BCV a source volume R1, and, using SRDF, links it to target volume R2 in SYMM6. Thus, the data which was copied to R1BCV from R1 in SYMM2 can now be cascaded down to target volume R2 in SYMM6.

With the present invention, host component software **08** can issue a query command (a query command allows the user to get host component software **08** to determine the status of any BCV volume) with remote parameters (as shown in FIG. **5** and described above) to locate all or a specified number of the BCV volumes at the remote sites related to the logical volume in question. Since the SRDF features in the SYMMETRIX™ systems already maintain the information about the remote volumes inside the SYMMETRIX™ systems themselves, in a table such as the illustrative Table in FIG. **16**, the remote parameters identify sets of remotely linked devices the user wishes to know about.

FIG. **2** illustrates the way in which the BCV features operate with an SRDF source volume R1, on SYMM1 using the present invention. As seen here, an establish BCV command issued for source device R1, will result in the establishment of a BCV volume on SYMM1 of host **02**. Normally, the data for the BCV volume will be copied from the source volume R1 on SYMM1, unless problems necessitate obtaining the data from the target volume R2 device in the SYMM3 device attached to host **04**. If the BCV device is split from the source device, R1, this does not affect data transfers between source volume R1 and target volume R2. For restore and incremental restore commands, if the source volume R1/target volume R2 logical device link is suspended, the restore or incremental restore is only made locally to the source R1 device and the information about changed tracks is retained for later synchronization with the target volume R2.

Turning now to FIGS. **3***a* and **3***b*, the present invention also enables BCV volume activities at remote locations to be controlled through cascading commands issued from a host **02**. Establish and reestablish processes are the same as those described in application 020 if BCV devices are used in conjunction with an target volume R2 as the standard device.

Turning now to FIG. **3***a*, the present invention allows a user at host **02** to establish a business continuance volume BCV at a remote CAMPUS location, linked by SRDF to SYMMETRIX™ system SYMM3. In this approach, the SRDF functions have already linked source volume R1 of SYMM1 to target volume R2 of SYMM3, by assigning it as a remote mirror, thus assuring transparent copying of the contents of R1 to R2. As noted above, in the discussion of the SRDF features, this copying is done in the background by the SRDF features of SYMM3 and does not take time away from normal processing at host **02**. Similarly, the copying done to create a BCV volume is also done in the background in SYMM3, using the BCV features generally described in co-pending applications 953 and 020, without taking time away from normal processing in host **02** or host **04**.

The unique aspect shown in FIG. **3***a* is that the user can intervene at a remote SYMMETRIX™ system SYMM3, using the cascading commands of the present invention, to establish a BCV volume there. If a split command is subsequently done, the SYMMETRIX™ unit containing the target volume R2 device associated with the BCV pair locks the target volume from further updates and suspends the remotely mirrored R1 and R2 pair for a few seconds to execute the split. When the split is complete the link is restored, any changed tracks from source volume R1 are propagated to the target volume R2 for synchronization purposes. In this example, after the split, a BCV device containing a copy of the standard target volume R2 data is now available for use by host **04**.

In FIG. **3***b*, another aspect of the present invention is shown. Since the SRDF features of SYMMETRIX™ sys-

tems can also be bidirectional, as well as unidirectional, SYMM1 can be configured by SRDF functions to contain a target volume R2, which is a remote mirror of source volume R1 in SYMM3. As host **04** updates source volume R1, in SYMM3 its contents are automatically copied by SRDF features to target volume R2 in SYMM1. In this example, the present invention enables a user at host **02**, to establish a BCV volume in SYMM1 from target volume R2 in SYMM1. As mentioned above, a split command can then be used to stop the copying operation from R2 to the BCV volume, and frees up the new copy of the data on the BCV volume for other purposes, such as testing or restore operations. Effecting a split command does not affect SRDF data transfers with the target R2 device.

Now referring to FIG. *6a*, an establish with remote mirrors is shown. In FIG. *6a*, data volumes DV**00** and DV**01** are depicted as they are seen by the host. When these are established as a BCV pair, source volume BCVDV**01** R1 is assigned as the next mirror (M3) to standard volume SDV**00** and data from SDV**00** M1 and SDV**00** M2 is copied to BCVDV**01** R1, synchronizing it. Since, in this example, BCVDV**01** R1 is also a source volume in an SRDF link, it will be synchronized with target volume BCVDV**01** R2. In many cases, a BCV pair may be split and re-established a number of times.

The present invention provides that the user may request a differential split, as shown in FIG. *6b*, which splits off BCVDV**01** R1 from the standard volumes, and enables copying of only the updated tracks on BCVDV**01** R1 to its remote mirror BCVDV**01** R2.

FIG. *6c* illustrates a reverse split. A reverse split resynchronizes the BCV volume, here BCVDV**01** R1 with a full data copy from its mirror—here remote target volume BCVDV**01** R2. Prior to the present invention, without a reverse split command, a user could not put a BCV volume back to its original state if there had been any writes to the standard volume while the BCV volume was attached. When doing a restore, from a BCV device to a standard device no I/O commands could be sent to the standard device, since they would be propagated to the BCV device, thus destroying its integrity before the restore was complete. Heretofore, if a user wanted to preserve the data contents of the BCV device, the user had to wait for a restore to complete, and then issue a split command. When the restore completed, I/O's to the standard device could resume. Since a restore might take anywhere from 1–15 or more minutes, or might not be scheduled for hours, keeping the integrity of the data contents of the BCV device could be costly for performance of the system using the standard device. Yet, frequently, BCV devices are used to capture the status of data as of a certain day and time, so testing can be done against this copy of the real data. If the systems using the test data have bugs, it is often easier to find them by using the same data for testing until all bugs are fixed. Thus, for some applications, maintaining the integrity of the BCV data is important.

With the reverse split of the present invention, as soon as a reverse split is initiated, a user can immediately start using the standard device of the BCV pair, and send I/O operations to it. This is because the copying in a reverse split goes from the BCV device's mirror, remote mirror BCVDV**01** R2 in this example, to the BCV device BCVDV**01** R1, thus keeping the BCV data intact. In a reverse split, a full copy of the data is transmitted from the remote (in this example) mirror.

Turning now to FIG. *6d*, a reverse differential split is illustrated. Here, only the out-of-sync tracks are copied (instead of a full copy) from remote mirror BCVDV**01** R2 (in this example) to the BCV device BCVDV**01**.

Thus, the use of the reverse split and reverse differential split allows a user to keep the BCV data contents as of a certain date and time unchanged on the remote mirror (in these examples), and use them to return a changed BCV volume to the original contents of that date and time. All of this can be done without interrupting processing on the standard volumes.

Returning now to FIG. **1**, it can be seen that if restore operations were the only way to restore BCV data from a remote site, such as SYMM**6**, the delay could be further compounded by telecommunications delays over the SRDF EXTENDED DISTANCE link. With the reverse differential split, a changed BCV volume can be brought to the status of the BCV data as of a certain date with no interruption of I/O to the standard devices.

Still in FIG. **1**, SYMMETRIX™ systems capable of implementing the cascading commands of the present invention include microcode in the remote link directors and other parts of the SYMMETRIX™ system, which gives a host attached to the SYMMETRIX™ system at a source location, the ability to issue cascading BCV commands across SRDF links to devices in the SYMMETRIX™ at target locations. These cascading BCV commands are pass-through commands that travel across a link defined by an RLD as an R1/R2 pair, such as that shown in SYMM1 and SYMM2. The commands can also be passed through a configured socket device that is a member of the specified remote access (RA) group. In FIG. **1**, this is exemplified at SYMM5 and SYMM6, where SRDF communications occur over a telecommunications link RA**0**. In FIG. **16**, the microcode tables kept by each SRDF configured SYMMETRIX™ system are modified in accord with the present invention to include an additional address indicator—line**1**, addr—for indicating the place of this unit in a chain. This address indicator can be as simple as additional bytes to include more references, or as complex as a vector which points to other addresses.

Turning now to FIG. **15**, a flow diagram of the present invention is shown. Since most of the processing for BCV volume commands, such as establish, split, query, re-establish, restore, etc., are diagrammed in application 020, referenced above, this flow diagram concentrates on the general flow followed for parsing and acting upon remote, cascading commands—block **150**. At decision block **155**, the present invention checks the command to see whether the RMT parameter has been specified. If it has not, local processing continues at step **160**. If it has been specified, the present invention parses the command syntax (illustrated in FIGS. *4a–4e* and FIG. **5**) to identify the remote access group ragrp and remote devices which are the targets of the command. As noted above in the discussion of FIG. **5**, this command syntax uses a simple positional remote access group number to identify the path. The invention checks at step **175** to see if an ragrp has been specified. If it has not, at step **180** the system checks to see if the device specified is a source volume, R1. If it is not, then this is an error in a remote command. If the device is an R1, the invention proceeds to step **200**, and uses the ragrp number that the R1 is associated with, and proceeds to step **170**. If an ragrp was specified or has been assigned as in step **200**, at step **170**, a path is defined and the pointer to the first ragrp link in it is indicated. At that point, the present invention returns to normal command processing. When a remote cascading command is issued, it follows the link paths defined for it by the user, and the present invention. Thus, a command can take effect several "hops" away from the local system.

While the present invention applies these cascading commands to BCV commands, those skilled in the art will appreciate that the same apparatus and method could be used for other commands designed to take effect at a distance through several intervening systems.

Those skilled in the art will appreciate that while a preferred embodiment of the present invention is directed to handling mass storage disk volumes connected to host systems, other types of writable storage media requiring remote access could also be handled by the method and apparatus of the present invention. Similarly, while the remote commands of the present system are illustrated primarily for use with business continuance volumes, those skilled in the art will appreciate that they could apply to other types of activity in a remote system setup.

In this embodiment, host component software **08**'s implementation of the remote control cascading commands is written in assembler language for IBM mainframes using the MVS operating system. Those skilled in the art will appreciate that it could also be written in other languages, such as C or C++, for example. Similarly, while the present invention is designed for use with the MVS operating system, those skilled in the art will appreciate that variations of it could be implemented for other operating systems such as UNIX™, NT™, and others without deviating from the spirit of the present invention.

Those skilled in the art will appreciate that the embodiments described above are illustrative only, and that other systems in the spirit of the teachings herein fall within the scope of the invention.

What is claimed is:

**1**. In a system comprising a local mass storage system which is linked to a host system and to a physically remote mass storage system, the physically remote mass storage system being linked to one or more other physically remote mass storage systems so as to form a stream of mass storage systems, each pair of mass storage systems being linked by a communications path and a pair of remote link directors, each of the mass storage systems comprising one of the pair of remote link directors, and each such mass storage system further comprising a cache memory, a disk adapter and one or more disk drives, a host system providing local control of mass storage volumes located on the local mass storage system and remote control of linked mass storage volumes located on any of the remote mass storage systems in the stream of mass storage systems, comprising:

a mechanism at the host system that generates and issues a user-specified command from the host system to a local mass storage system;

a selected identifier being in the command as issued from the host system, the command being recognizable by each mass storage system in the stream of mass storage systems, the selected identifier being selected from a group of identifiers, the identifiers in the group identifying the mass storage systems in the stream of mass storage systems, each mass storage system that is not identified by the selected identifier being configured to forward the command to another mass storage system, at least one mass storage system that is identified by the selected identifier being configured to execute the command after the at least one mass storage system receives the command,

a plurality of user-specifiable parameters also being specifiable in the command as issued from the host system, the parameters that may be specified in the command including a first parameter, a second parameter, and one or more third parameters, the presence of the first parameter in the command indicating that the command is to be executed physically remotely from the local mass storage system, the second parameter indicating an order in which the command is to be executed, and the one or more third parameters indicating one or more paths via which the command is to be forwarded to the at least one mass storage system, the absence of the first parameter from the command indicating that the command is to be executed at the local mass storage system.

**2**. The host system of claim **1**, further comprising host component software for processing command syntax to allow a user to specify business continuance operations to be effected on the at least one mass storage system that is identified by the selected identifier.

**3**. The host system of claim **2**, wherein said business continuance operations comprise an establish command for causing said at least one mass storage system that is identified by the selected identifier to join a specified standard volume at a local site and a specified business continuance source volume into a business continuance pair by copying contents of said standard volume onto said business continuance source volume and a target volume mirror at the at least one mass storage system.

**4**. The host system of claim **3**, wherein said business continuance operations further comprise a split command for causing said at least one mass storage system to separate said business continuance pair and flag a write to said standard volume and said business continuance source volume which may occur after said split.

**5**. The host system of claim **4**, wherein said business continuance operations further comprise a reverse split command for causing said at least one mass storage system to separate said business continuance pair and copy all the contents from the business continuance volume to the associated source volume.

**6**. The host system of claim **4**, wherein said business continuance operations further comprise a reverse differential split command for causing said at least one mass storage system to separate said business continuance pair and copy only changed contents from the business continuance target volume to the associated source volume.

**7**. The host system of claim **2**, wherein said business continuance operations comprise a re-establish command for causing said at least one mass storage system to rejoin a standard volume and a business continuance source volume which had once been a business continuance pair into another business continuance pair, by copying to said business continuance source volume writes to said standard volume which had been flagged after a split and copying from said standard volume any data which had been changed by writes to said business continuance source volume which had been flagged after said split.

**8**. The host system of claim **2**, wherein said business continuance operations comprise a restore command for causing said at least one mass storage system to copy contents of a business continuance source volume of a business continuance pair to a standard volume.

**9**. The host system of claim **2**, wherein said business continuance operations comprise an incremental restore command for causing said at least one mass storage system to copy to a standard volume writes to a business continuance source volume which have been flagged since said business continuance source volume was split from a business continuance pair.

**10**. The host system of claim **2**, wherein said business continuance operations comprise a differential split com-

mand for causing said at least one mass storage system to separate a business continuance pair and copy only changed data from the business continuance source volume to a target volume mirror.

11. In a system comprising a local mass storage system which is linked to a host system and to a physically remote mass storage system, the physically remote mass storage system being linked to one or more other physically remote mass storage systems so as to form a stream of mass storage systems, each pair of mass storage systems being linked by a communications path and a pair of remote link directors, each of the mass storage systems comprising one of the pair of remote link directors, and each such mass storage system further comprising a cache memory, a disk adapter and one or more disk drives, a method of enabling a host system to perform local control of mass storage volumes located on the local mass storage system and remote control of linked mass storage volumes located on any of the remote mass storage systems in the stream of mass storage systems comprising the steps of:

issuing a user-specified command from the host system to the local mass storage system; and

placing at the host system a selected identifier in the command, the command being recognizable by each mass storage system in the stream of mass storage systems, the selected identifier being selected from a group of identifiers, the identifiers in the group identifying the mass storage systems in the stream of mass storage systems, each mass storage system that is not identified by the selected identifier being configured to forward the command to another mass storage system, at least one mass storage device that is identified by the selected identifier executing the command after the at least one mass storage system receives the command,

a plurality of user-specifiable parameters also being specifiable in the command as issued from the host system, the parameters that may be specified in the command including a first parameter, a second parameter and one or more third parameters, the presence of the first parameter in the command indicating that the command is to be executed physically remotely from the local storage system, the second parameter indicating an order in which the command is to be executed, and the one or more third parameters indicating one or more paths via which the command is to be forwarded to the at least one mass storage system, the absence of the first parameter from the command indicating that the command is to be executed at the local mass storage system.

12. The method of claim 11, wherein said method further comprises the step of using host component software for processing command syntax to allow a user to specify business continuance operations to be effected on the at least one mass storage system that is identified by the selected identifier.

13. The method of claim 12, wherein said business continuance operations comprise the step of using an establish command for causing said at least one mass storage system that is identified by the selected identifier to join a specified standard volume at a local site and a specified business continuance source volume into a business continuance pair by copying contents of said standard volume onto said business continuance source volume and a target volume mirror.

14. The method of claim 13, wherein said business continuance operations further comprise the step of using a split command for causing said at least one mass storage system to separate said business continuance pair and flag a write to said standard volume and said business continuance source volume which may occur after said split.

15. The method of claim 14, wherein said business continuance operations further comprise the step of using a reestablish command for causing said at least one mass storage system to rejoin a standard volume and a business continuance source volume which had once been a business continuance pair into the business continuance pair again, by copying to said business continuance source volume said writes to said standard volume which had been flagged after said split and copying from said standard volume any data which had been changed by said writes to said business continuance source volume which had been flagged after said split.

16. The method of claim 14, wherein said business continuance operations further comprise the step of using a restore command for causing said at least one mass storage system to copy contents of a business continuance source volume of a business continuance pair to a standard volume.

17. The method of claim 14, wherein said business continuance operations further comprise the step of using an incremental restore command for causing said at least one mass storage system to copy to said standard volume said writes to said business continuance source volume which have been flagged since said business continuance source volume was split from said business continuance pair.

18. The method of claim 14, wherein said business continuance operations further comprise the step of using a differential split command for causing said at least one mass storage system to separate said business continuance pair and copy only changed data from the business continuance source volume to a target volume mirror.

19. The method of claim 14, wherein said business continuance operations further comprise the step of using a reverse split command for causing said at least one mass storage system to separate said business continuance pair and copy all the contents from a business continuance volume to an associated source volume.

20. The method of claim 14, wherein said business continuance operations further comprise the step of using a reverse differential split command for causing said at least one mass storage system to separate said business continuance pair and copy only changed contents from a business continuance target volume to an associated source volume.

* * * * *

# United States Patent [19]

**Shimizu et al.**

[11] **Patent Number: 4,985,828**

[45] **Date of Patent: Jan. 15, 1991**


[54] **METHOD AND APPARATUS FOR GENERATING A REAL ADDRESS MULTIPLE VIRTUAL ADDRESS SPACES OF A STORAGE**


[75] Inventors: **Naohiko Shimizu; Hideo Sawamoto**, both of Hadano, Japan

[73] Assignee: **Hitachi, Ltd.**, Tokyo, Japan



[21] Appl. No.: **156,454**

[22] Filed: **Feb. 16, 1988**

[30] **Foreign Application Priority Data**

Mar. 19, 1987 [JP] Japan ..... 62-65424

[51] **Int. Cl.**[5] ..... **G06F 12/10**

[52] **U.S. Cl.** ..... **364/200**; 364/900; 364/256.3; 364/961.2; 364/246.6

[58] **Field of Search** ... 364/200 MS File, 900 MS File, 364/134



[56] **References Cited**

U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 4,004,278 | 1/1977 | Nagashima | 364/200 |
| 4,136,385 | 1/1979 | Gannon et al. | 364/200 |
| 4,326,248 | 4/1982 | Hinai et al. | 364/200 |
| 4,355,355 | 10/1982 | Butwell et al. | 364/200 |
| 4,521,846 | 6/1985 | Scalzi et al. | 364/200 |
| 4,758,951 | 7/1988 | Sznyter, III | 364/200 |
| 4,761,737 | 8/1988 | Duvall et al. | 364/200 |
| 4,812,969 | 3/1989 | Takagi et al. | 364/200 |
| 4,843,541 | 6/1989 | Bean et al. | 364/200 |

FOREIGN PATENT DOCUMENTS

| | | |
|---|---|---|
| 60-48494 | 12/1984 | Japan . |
| 60-68443 | 4/1985 | Japan . |



*Primary Examiner*—Michael R. Fleming
*Assistant Examiner*—Gopal C. Ray
*Attorney, Agent, or Firm*—Antonelli, Terry, Stout & Kraus



[57] **ABSTRACT**

A multiple virtual space control in a multiple virtual storage system having an address translation table used to translate a logical address to a real address, a control register for holding a start address of the address translation table or a space identifier (hereinafter represented by address translation table start address) and an address translation buffer containing a pair of logical address and real address and an address translation table start address for translating a logical address to a real address, in order to update the content of the control register to switch the virtual space. A group identifier comprising a plurality of bits for identifying an area common to a group of virtual spaces is added to an entry of the address translation table, an entry of the address translation buffer and the control register. When a logical address is to be translated to a real address, if there is an entry having a logical address and an address translation table start address equal to the memory request logical address and the address translation table start address of the control register, or an entry having a logical address and a group identifier equal to the memory request logical address and the group identifier of the control register, in the address translation buffer, the real address of the entry is rendered valid and used for memory access.


**14 Claims, 1 Drawing Sheet**

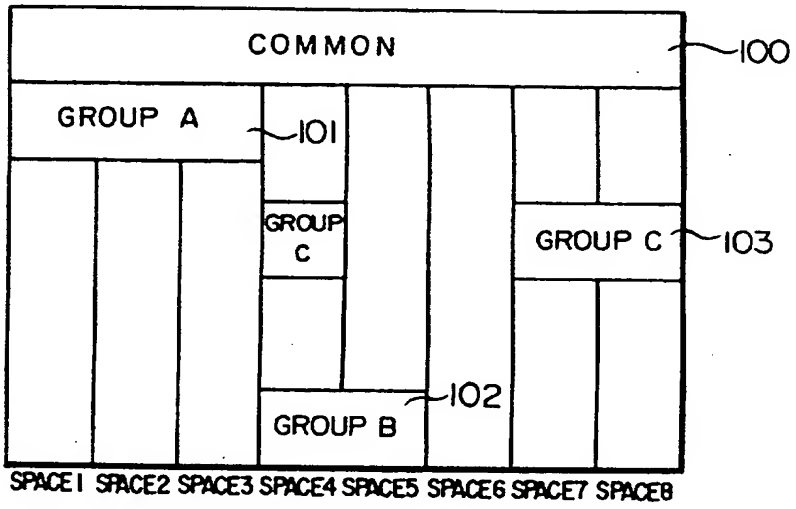


FIG. 1

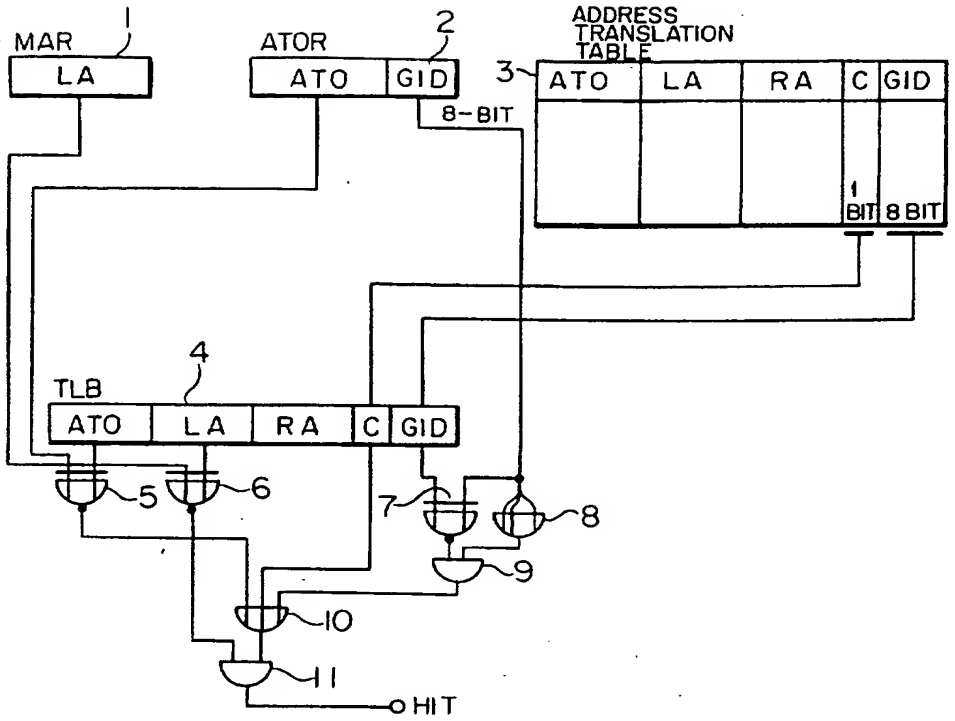
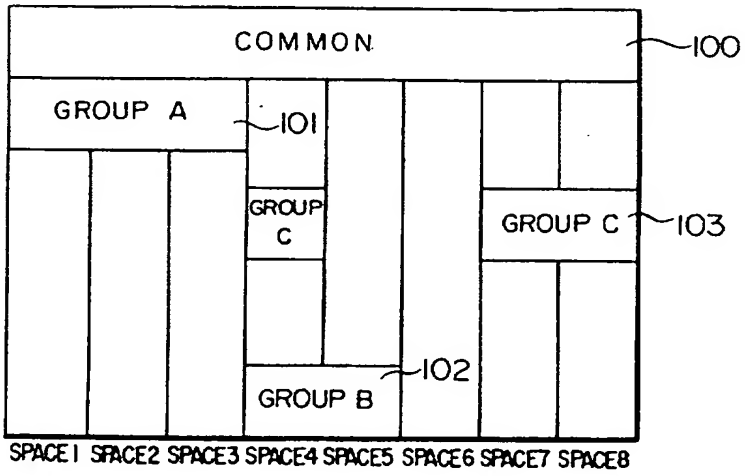

MAR
1
LA
ATOR
2
ATO
GID
8-BIT
ADDRESS TRANSLATION TABLE
3
ATO
LA
RA
C
GID
1 BIT
8 BIT
TLB
4
ATO
LA
RA
C
GID
5
6
7
8
9
10
11
HIT

FIG. 2


COMMON
100
GROUP A
101
GROUP C
GROUP C
103
102
GROUP B
SPACE1 SPACE2 SPACE3 SPACE4 SPACE5 SPACE6 SPACE7 SPACE8

# METHOD AND APPARATUS FOR GENERATING A REAL ADDRESS MULTIPLE VIRTUAL ADDRESS SPACES OF A STORAGE

## BACKGROUND OF THE INVENTION

The present invention relates to method and apparatus for translating an address of a computer system which adopts a virtual memory scheme, and more particularly to method and apparatus for controlling multiple virtual spaces suitable for address translation where there is a common area among a portion of the virtual address space in a multiple virtual storage system.

In the computer system of the virtual memory scheme, a memory address is given by an address on a virtual space (that is, a logical address). Accordingly, when a main storage is to be accessed, it is necessary to translate the logical address to an address on the main storage (that is, a real address). The address translation is carried out by looking up an address translation table (usually comprising a segment table and page tables) provided in the main storage. Pairs of logical addresses and real addresses obtained by the address translation are usually registered or stored in an address translation buffer or translation lookaside buffer (TLB), and for the logical address requested for the memory access, the corresponding real address is looked up from the TLB so that the address translation is carried out at a high speed.

On the other hand, in a multi-process multiple virtual storage system, a virtual space is allotted for each job and an address translation table is provided for each virtual space. Switching of virtual spaces due to switching of jobs is carried out under control of an operating system (OS) of a computer system, such as Hitachi VOS3, by rewriting a start address of the address translation table or rewriting a content of an address control register (ATOR) which holds a space identifier. In this case, in addition to the logical address (LA) and the real address (RA), the content (ATO) of the address control register are held in the TLB. When the object real address is to be looked up from the TLB, the coincidence of the corresponding virtual spaces as well as the coincidence of the logical addresses is checked.

In such a multiple virtual storage system, a common area to respective virtual spaces is indicated by adding a common segment bit (C bit) in the TLB. When the C bit is "1" (indicating the presence of the common area), the start address of the address translation table or the space identifier held in the TLB is ignored, and if the logical address from the memory address register and the logical address of the TLB are equal, the corresponding real address is considered valid and the accessing to the main storage is carried out thereby. In the multiple virtual computer system, a computer identifier (CN bit) is set to an area common to the virtual computers, and a pair consisting of a logical address and a real address registered in the TLB is rendered valid by a logical product (AND) condition of the C bit and the CN bit.

This type of virtual space control system is shown in JP-A-60-68443 filed by the assignee of the present application and published on Apr. 19, 1985.

The prior art described above is effective in enhancing a hit rate of the address translation buffer when the common area which is common to all virtual spaces is used, but it does not give consideration to a case where there is a common area among a portion of virtual spaces. When the common area is set for the portion of virtual spaces, the hit rate of the address translation buffer decreases and the number of times of rewriting of the buffer increases. Data in the common area can be readily read from or updated by any virtual space so that security for the data and programs of the respective virtual spaces is not sufficiently assured. Accordingly, the prior art system is hardly applicable to an area which is to be shared by only specified spaces.

## SUMMARY OF THE INVENTION

It is an object of the present invention to provide a method and apparatus for controlling address translation of multiple virtual spaces of a multi-process machine which is suitable when a partial common area is shared by a plurality of virtual spaces.

It is another object of the present invention to provide a method and apparatus for controlling address translation of multiple virtual spaces which permits setting of a common area without resulting in decrease of a hit rate of a TLB among specified virtual spaces and which can assure security of data and programs.

In order to achieve the above objects, in accordance with the present invention, a group identifier comprising a plurality of bits for identifying a group of virtual spaces having a partial common area is added to each of entries of an address translation table, entries of an address translation buffer and a control register which holds a start address of the address translation table or a space identifier, and when there is an address translation buffer entry having a coincidence between a virtual address from a memory address register and the start address of the address translation table or the space identifier from the address control register, or there is an address translation buffer entry having a coincidence between the virtual address (LA) from the memory address register and the group identifier (GID) from the address control register, the real address of the address translation buffer entry is considered valid and the main storage is accessed by that real address.

When address translation is carried out for one virtual space under the control of the operating system (OS) of the virtual memory scheme, a virtual address and a real address as well as a current start address of the address translation table or a space identifier and a group identifier added to the entry of the corresponding address translation table are registered into the entry of the address translation buffer. The group identifier comprises a plurality of bits so that it can identify a plurality of predetermined groups of virtual spaces.

In the address translation in other virtual spaces having the same group identifier, if there is an entry having a coincidence between a requested virtual address and the virtual address in the address translation buffer and a coincidence between a group identifier from the address control register and the group identifier in the address translation buffer, the address translation is carried out by the address translation buffer and no new registration is made to the address translation buffer. Accordingly, rewriting of the buffer is not necessary. In the address translation in other virtual spaces having a different group identifier, there is no coincidence between group identifiers nor the start addresses of the address translation tables or space identifiers. Accordingly, the address translation is carried out by looking up the address translation table and the result is registered into the address translation buffer. Thus, even if

the virtual addresses are equal, different address translation tables are looked up.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram showing one embodiment of a system configuration of the present invention, and

FIG. 2 is a diagram showing an example of a memory map of multiple virtual spaces considered in the present invention.

## DESCRIPTION OF THE PREFERRED EMBODIMENTS

One embodiment of the present invention is now explained with reference to the accompanying drawings.

FIG. 2 shows a memory map of multiple virtual spaces considered in the present invention. In FIG. 2, each of the virtual spaces 1-8 has an area 100 (wholly common area) which is common to all virtual spaces 1-8, as described in the above-referenced JP-A-60-68443. In the present embodiment, the virtual spaces 1-3 have a partial common area 101 (Group A), the virtual spaces 4 and 5 have a partial common area 102 (Group B), and the virtual spaces 4, 7 and 8 have a partial common area 103 (Group C). Hereinafter, an identifier for the wholly common area 100 is called a common segment bit (C), and identifiers for the Groups A, B and C (101-103) are called group ID's (GID's). The GID may comprises eight bits with respective bits corresponding to the spaces 1-8. For the Group A, the bits 1-3 are "1", for the Group B, the bits 4 and 5 are "1", and for the Group C, the bits 4, 7 and 8 are "1". The Groups A, B and C may be identified by codes. In this case, GID may comprise less than eight bits.

FIG. 1 shows a block diagram of one embodiment of an address translation control system of the present invention. It shows a hit detector of an address translation buffer.

In FIG. 1, a memory address register (MAR) 1 holds a logical address (LA) of a memory access request. An address translation table start address register (ATOR) 2 holds a space identifier or an address translation table start address (ATO) and a group ID (GID) in order to identify a current virtual space. When a virtual space is selected and switched under control of an OS, the space identifier or ATO in the ATOR 2 is updated. In the following description, either the space identifier or the ATO is used in the same manner.

Each entry of the address translation table 3 comprises a real address (RA), a common segment bit (C) and a group ID (GID). The address translation table 3 is provided in a main storage for each virtual space, and a start address thereof is designated by the ATO in the ATOR 2. The address translation table usually comprises a segment table and page tables, but they are shown as one table in the present embodiment for the purpose of simplification.

Each entry of the address translation buffer (TLB) 4 comprises an address translation table start address (ATO), a logical address (LA), a real address (RA), a common segment bit (C) and a group ID (GID). While only one entry is shown, the TLB 4 actually has a plurality of entries and a desired entry is accessed by a predetermined bit such as a start address of the memory address register (MAR) 1. This method is referred to as a set associative address translation buffer control. When the address translation is carried out by using the address translation table 3, the logical address (LA) of the memory address register 1, the ATO (or space identifier) of the ATOR 2, and RA, C and GID of the address translation table 3 are registered into the corresponding entry of the TLB 4.

An exclusive OR circuit (EX-OR) 5 compares the ATO of the ATOR 2 and the ATO of the TLB 4 and produces a logical "1" output when they are equal. An EX-OR 6 compares the logical address of the memory address register 1 and the LA of the TLB 4 and produces a logical "1" output when they are equal. An EX-OR 7 compares the GID of the ATOR 2 and the GID of the TLB 4 and produces a logical "1" output when they are equal. Those EX-OR's produce logical "0" signals when compared inputs are not equal. An OR circuit 8 constitutes detection means for detecting if a virtual space designated by the OS does not belong to any common group and produces a logical "1" output when at least one of the bits of the GID of the ATOR 2 is a significant bit and an input logical signal is "1". An AND circuit 9 produces a logical "1" output when the outputs of both the EX-OR 7 and the OR circuit 8 are "1". An OR circuit 10 produces a logical "1" output when one of the output of the EX-OR 5, the C bit of the TLB 4 and the output of the AND circuit 9 is "1". An AND circuit 11 produces a logical "1" output when the outputs of both of the EX-OR 6 and the OR circuit 10 are "1". The output of the AND circuit 11 is an address translation buffer hit signal HIT. When the signal HIT is "1", the RA of the TLB 4 is rendered valid and the memory is accessed by that real address.

The operation of FIG. 1 is explained with reference to the memory map shown in FIG. 2.

It is assumed that values of the area 101 of the Group A of the virtual space 1 have been registered in the ATO, LA, RA, C and GID of the entries of the TLB 4, and the ATO and GID of the virtual space 2 have been set in the ATOR 2. If the logical address of the memory address register MAR 1 designates an area of the Group A, the EX-OR 6 produces a logical "1" output. Since the ATO of the ATOR 2 and the ATO of the TLB 4 are now not equal, the EX-OR produces a logical "0" output. Since the area 101 of the Group A is not the wholly common area 100, the C bit of the TLB is "0". In this case, if the group ID (GID) has not been set, the OR circuit 10 produces a logical "0" output because the circuits 7-9 produce logical "0" outputs and the AND circuit 11 produces a logical "0" output which indicates that the corresponding logical address does not exist in the TLB (not in TLB).

On the other hand, if values indicating the common Group A to the virtual spaces 1 and 2 (that is, bits 1-3 of the GID are "1") have been set in the GID of the ATOR 2, the outputs of both the EX-OR 7 and the OR 8 are "1" and the OR circuit 10 produces a logical "1" output. As a result, the HIT signal of the AND circuit 11 is "1" indicating that a real address corresponding to the logical address exists in the TLB (in TLB). Thus, the main memory is accessed by using the RA of the TLB 4 as the real address. It is noted that the space 4 included in Group B and Group C is relatively small in the number of reference thereto.

In accordance with the present invention, the address translation can be attained by the same address translation buffer entry for the partially common area in different virtual spaces. Thus, in the set associative address translation buffer control, the number of times of updating of the address translation buffer decreases and the buffer hit rate is enhanced. Further, the accessing to a

common area of a group of virtual spaces by a virtual space of another group can be inhibited by setting different group ID's for the respective groups.

We claim:

1. A method of generating a real address in multiple virtual address spaces of a storage having multiple virtual storage spaces, an address translation table for each virtual address space to translate a logical address to a real address, an address control register for storing a start address of the address translation table or a virtual address space identifier, and an address translation buffer capable of storing copies of entries contained in the address translation table, said method comprising the steps of:

providing group identifiers in each of the entries in said address translation table, said address translation buffer and the entry in said address control register for identifying a respective area shared by a specific group of virtual spaces;

determining whether there is at least one entry in the address translation buffer whose logical address and address translation table start address conform with the logical address supplied for a memory request and the address translation table start address in said address control register, or whether an entry exists in the address translation buffer whose logical address and group identifier conform to the logical address supplied for memory access and the group identifier of said address control register; and

if there is at least one conformity, identifying the real address of said entry as a valid real address for memory access.

2. A method according to claim 1 wherein the shared areas include a plurality of different areas including a predetermined combination of portions of virtual address spaces smaller in number than the total number of virtual spaces, each of said different areas having a group identifier for identifying the particular area, said areas being independent from one another and non-accessible by each other.

3. A method according to claim 1 wherein the shared areas include an area common to at least one first and one second virtual address space, each virtual address space having a plurality of limited common sub-areas, and data in the common sub-areas being managed by an operating system.

4. A method according to claim 3 wherein said first virtual address space has a higher priority than said second virtual address space.

5. A method according to claim 1 wherein said determining step includes sub-steps of (1) comparing the logical address supplied from a memory address register with the logical address in said address translation buffer to produce a first coincidence signal when they are equal, and comparing the group identifier of said address control register and the group identifier of said address translation buffer to produce a second coincidence signal when they are equal, and (2) determining that the corresponding entry in said address translation buffer is valid when said first coincidence signal and said second coincidence signal are simultaneously present.

6. A method according to claim 5 wherein said second coinciding signal is inhibited when it is detected that the group identifier from said address control register does not belong to any group of virtual address spaces with shared areas.

7. A method according to claim 1 further comprising a step of copying the entry in the address translation table into said address translation buffer if it is found that the corresponding entry does not exist in the address translation buffer.

8. A virtual address space control apparatus in a multiple virtual storage system, comprising:

(a) an address translation table having entries each including a pair of logical and real addresses and a group identifier for identifying storage areas common to specified groups of virtual address spaces, for translating a logical address to a real address under control of an operating system;

(b) an address translation buffer having logical and real address pairs copied from the address translation table, and an associated group identifier and an associated virtual address space identifier for translating a logical address to a real address;

(c) an address control register specifying a virtual address space identifier and a group identifier; and

(d) detection means responsive to supply of a logical address for memory access for detecting the presence of entries in the address translation buffer of the kind whose logical address and virtual address space identifier are equal to the logical address supplied for memory access and the virtual address space identifier of said address control register, or for detecting entries of the kind whose logical address and group identifier are equal to the logical address supplied for memory access and the group identifier of said address control register;

said detection means, when the presence of either of said entries is detected, producing a signal for rendering the real address of said address translation buffer valid as a memory accessing address.

9. A control apparatus according to claim 8 wherein said group identifier includes a plurality of bits, while a single bit identifying a common area to all virtual address spaces is provided in said address translation table and in the address translation buffer.

10. A control apparatus according to claim 9 wherein said detection means include means for comparing the logical address supplied for a memory access from a memory address register and the content of the corresponding entry of said address translation buffer to produce a first coincidence signal when they are equal, means for comparing the group identifier of said address control register and the content of the corresponding entry of said address translation buffer to produce a second coincidence signal when they are equal, and select signal producing means responsive to simultaneous occurrence of said first coincidence signal and said second coincidence signal for producing a signal to select the real address of the corresponding entry in said address translation buffer as an address for memory access.

11. A control apparatus according to claim 10 wherein said means for producing the second coincidence signal includes means for inhibiting the second coincidence signal when it is detected that the group identifier from said address control register does not belong to any group of virtual address spaces having common areas.

12. A control apparatus according to claim 10 wherein said detection means further includes means for comparing the control of the output from said address control register and the content of the corresponding entry of said address translation buffer to produce a

third coincidence signal when they are equal, said select signal producing means producing a select signal upon the simultaneous occurrence of said first coincidence signal and said third coincidence signal.

13. A control apparatus according to claim 12 wherein said means for producing a select signal responds to simultaneous occurrence of a signal indicating the common area to all virtual address spaces in said address translation buffer and said first coincidence signal.

14. A control apparatus according to claim 8 wherein said virtual address space identifier comprises a start address of said address translation table.

* * * * *